

RHCE Rapid Track Course
Student Workbook
Red Hat Enterprise Linux 6
Release 2-20101223





RHCE RAPID TRACK COURSE

Red Hat Enterprise Linux 6 RH300

RHCE Rapid Track Course

Edition 2

Author	Forrest Taylor
Author	David Duffey
Author	George Hacker
Author	Joshua Hoffman
Author	Robert Locke
Author	Bowe Strickland
Editor	Steven Bonneville
Editor	Mark Howson

Copyright © 2010 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2010 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Contributors: Brian Butler, Victor Costea, Andrew Dingman, Chris Negus

Document Conventions	vii
Notes and Warnings	vii
Introduction	ix
Welcome to class!	ix
About Red Hat Enterprise Linux	ix
Additional Red Hat Enterprise Linux Software	x
Contacting Red Hat Technical Support	xii
About This Course	xv
RHCE Rapid Track Course	xv
Structure of the Course	xv
Orientation to the Classroom Network	xvi
Internationalization	xvii
Language Support	xvii
System-wide Default Language	xvii
Per-user Language Selection	xvii
Input Methods	xviii
Language Codes Reference	xviii
1. Software Management	1
Register with Red Hat Network (RHN)	2
Using Third-Party Repositories	6
Using yum	10
Handling Third-Party Software	13
RPM Package Design	17
RPM Package Specifications	18
Building and Signing an RPM Package	24
Publish RPM Packages	29
Criterion Test	31
2. Network Management	35
Understanding Network Configuration Files	36
Network Troubleshooting Toolkit	40
Network Interface Configuration - IP Aliases	43
Network Interface Configuration - Bonding	45
Tuning Kernel Parameters	48
Criterion Test 1	51
Criterion Test 2	52
3. Storage Management	57
Simple Partitions and File Systems	58
Enabling Data Privacy with Partition Encryption	62
Managing Swap Space	65
Accessing iSCSI Storage	68
Criterion Test	71
4. Logical Volume Management	75
Recognize the Components of LVM	76
Implement LVM Storage with Command-line Tools	79
Extend a Logical Volume and ext4 Filesystem	82
Extending and Reducing a Volume Group	86
Create a Snapshot to Facilitate Data Backup	88

Criterion Test	90
5. Account Management	93
Managing Passwords	94
Managing Filesystem Access Control Lists	97
Criterion Test	101
6. Authentication Management	105
Network Authentication Using an LDAP Server	106
Kerberos Configuration	110
Troubleshooting System Security Services Daemon (SSSD)	112
Network Mounting Home Directories	114
Criterion Test	116
7. Installation, Kickstart and Virtualization	119
Creating a Kickstart File by Modifying a Template	120
Introduction to KVM Virtualization	124
Virtual Guest Installation	127
Manage Virtual Machines	129
Criterion Test	131
8. Boot Management	135
Resolve GRUB Issues	136
Making Persistent GRUB Changes	138
Changing the Default Run Level	140
Single-User Mode	142
The Boot Process and Rescue Mode	144
Repairing Boot Issues	151
Criterion Test	154
9. SELinux Management	157
Basic SELinux Security Concepts	158
SELinux Modes	161
Display and Modify SELinux Modes	164
Display and Modify SELinux File Contexts	166
Managing SELinux Booleans	169
Monitor SELinux Violations	171
Criterion Test	174
10. Firewall Management	177
Packet Filtering	178
Network Address Translation	183
Criterion Test	186
11. NTP Server Configuration	189
Configure an NTP Server	190
Criterion Test	194
12. System Logging Service	197
Usage Reports	198
Configure a Remote Logging Service	201
Criterion Test	205
13. Web Service	209
Securing Apache with Encryption	210

Configure Name-Based Virtual Hosting	213
Stage a CGI executable	216
Configure User-Based Authentication	219
Troubleshooting Apache SELinux issues	222
Criterion Test 1	226
Criterion Test 2	227
14. Basic SMTP Configuration	231
Basic E-mail Delivery	232
Intranet Configuration	235
Criterion Test	239
15. Caching-Only DNS Server	243
DNS Overview	244
Caching-only DNS Servers	247
Criterion Test	249
16. File Sharing with NFS	253
NFS Concepts and Configuration	254
Using NFS	259
Criterion Test	261
17. File Sharing with CIFS	265
Accessing CIFS Shares	266
Providing Home Directories as CIFS Shares	269
Configuring Group and Print CIFS Shares	274
Criterion Test	276
18. File Sharing with FTP	279
FTP Drop-box Anonymous Upload	280
Criterion Test	282
19. CUPS Service	285
Configure Printers	286
Manage Print Jobs	288
Criterion Test	290
20. SSH Service	293
Using SSH keys	294
Criterion Test	297
21. Virtual Network Computing (VNC) Service	301
Configuring a VNC Server	302
Secure access to a remote GNOME desktop	304
Criterion Test	306
22. Comprehensive Review	309
Comprehensive Review Test	310
A. Solutions	313
Software Management	313
Network Management	319
Storage Management	323
Logical Volume Management	327
Account Management	333
Authentication Management	336

Installation, Kickstart and Virtualization	341
Boot Management	350
SELinux Management	356
Firewall Management	360
NTP Server Configuration	365
System Logging Service	368
Web Service	371
Basic SMTP Configuration	381
Caching-Only DNS Server	387
File Sharing with NFS	389
File Sharing with CIFS	392
File Sharing with FTP	397
CUPS Service	399
SSH Service	401
Virtual Network Computing (VNC) Service	403
Comprehensive Review	406

Document Conventions

Notes and Warnings



Note

"Notes" are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



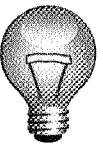
Comparison

"Comparisons" look at similarities and differences between the technology or topic being discussed and similar technologies or topics in other operating systems or environments.



References

"References" describe where to find external documentation relevant to a subject.



Important

"Important" boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled "Important" will not cause data loss, but may cause irritation and frustration.



Warning

"Warnings" should not be ignored. Ignoring warnings will most likely cause data loss.

Introduction

Welcome to class!

Thank you for attending this Red Hat training class. Please let us know if you have any special needs while at our training facility.

Please ask the instructor if you have any questions about the facility, such as operating hours of the facility and when you will have access to the classroom, locations of restrooms and break rooms, availability of telephones and network connectivity, and information about the local area.

As a courtesy to other students, please place your pager or cell phone's ringer on vibrate or mute, or turn off your devices during class. We ask that you only make calls during break periods.

If you have a personal emergency and are unable to attend or complete the class, please let us know. Thank you!

About Red Hat Enterprise Linux

This course is taught using Red Hat Enterprise Linux, an enterprise-targeted Linux distribution focused on mature open source software designed specifically for organizations using Linux in production settings.

Red Hat Enterprise Linux is sold on a subscription basis, where the subscription gives you continues access to all supported versions of the operating system in binary and source form, not just the latest one, including all updates and bug fixes. Extensive support services are included: a support contract and Update Module entitlement to Red Hat Network are included for the subscription period. Various Service Level Agreements are available that may provide up to 24x7 coverage with a guaranteed one hour response time for Severity 1 issues. Support will be available for up to seven years after a particular major release (ten years with the optional "Extended Update Support" Add-On).

Red Hat Enterprise Linux is released on a multi-year cycle between major releases. Minor updates to major releases are released roughly every six months during the lifecycle of the product. Systems certified on one minor update of a major release continue to be certified for future minor updates of the major release. A core set of shared libraries have APIs and ABIs which will be preserved between major releases. Many other shared libraries are provided, which have APIs and ABIs which are guaranteed within a major release (for all minor updates) but which are not guaranteed to be stable across major releases.

Red Hat Enterprise Linux is based on code developed by the open source community, which is often first packaged through the Red Hat sponsored, freely-available Fedora distribution (<http://fedoraproject.org/>). Red Hat then adds performance enhancements, intensive testing, and certification on products produced by top independent software and hardware vendors. Red Hat Enterprise Linux provides a high degree of standardization through its support for four processor architectures (32-bit Intel x86-compatible, AMD64/Intel 64 (x86-64), IBM POWER, and IBM mainframe on System z). Furthermore, we support the 4000+ ISV certifications on Red Hat Enterprise Linux whether the RHEL operating system those applications are using

is running on “bare metal”, in a virtual machine, as a software appliance, or in the cloud using technologies such as Amazon EC2.

Currently, the Red Hat Enterprise Linux product family includes:

- *Red Hat Enterprise Linux for Servers*: the datacenter platform for mission-critical servers running Red Hat Enterprise Linux. This product includes support for the largest x86-64 and x86-compatible servers and the highest levels of technical support, deployable on bare metal, as a guest on the major hypervisors, or in the cloud. Subscriptions are available with flexible guest entitlements of one, four, or unlimited guests per physical host. Pricing is based on the basis of the number of socket-pairs populated on the system motherboard, the number of guests supported, the level of support desired, and the length of subscription desired.

Red Hat Enterprise Linux for IBM POWER and *Red Hat Enterprise Linux for IBM System z* are similar variants intended for those system architectures.

- *Red Hat Enterprise Linux Desktop*: built for the administrator and end-user, Red Hat Enterprise Linux Desktop provides an attractive and highly productive environment for knowledge workers on desktops and laptops. Client installations can be finely tailored and locked down for simplicity and security for any workstation task.

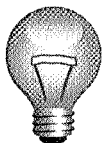
The basic *Desktop* variant is designed for task workers who have a limited amount of administrative control over the system, who primarily use productivity applications like Firefox Evolution/Thunderbird, OpenOffice.org, and Planner/TaskJuggler. The more sophisticated *Workstation* variant is designed for advanced Linux users who need a stand-alone development environment, and who are expected to have local super-user privileges or selected super-user privileges.

In addition, other variants exist such as *Red Hat Enterprise Linux for HPC Head Node* and *Red Hat Enterprise Linux for HPC Compute Node* (targeted at high-performance computing clusters), and *Red Hat Enterprise Linux for SAP Business Applications*. For more information please visit <http://www.redhat.com/>.

Additional Red Hat Enterprise Linux Software

Two additional software update channels are provided with Red Hat Enterprise Linux beyond the core software packages shipped:

- *Supplementary*: the "Supplementary" channel provides selected closed source packages, built for Red Hat Enterprise Linux as a convenience to the customer. These include things like Adobe Flash or proprietary Java JVMs.
- *Optional*: the "Optional" channel provides selected open source packages, as a convenience only. They are generally included in another Red Hat Enterprise Linux variant as a fully-supported package, or are a build requirement for the distribution. These packages are only available through a Red Hat Network child channel.



Important

Supplementary and *Optional* packages are provided with limited support, as a customer convenience only.

Red Hat also offers a portfolio of fully-supported *Add-Ons for Red Hat Enterprise Linux* which extend the features of your Red Hat Enterprise Linux subscription. These add-ons allow you to add capabilities and tailor your computing environment to your particular needs. These Add-Ons include support for high availability application clustering, cluster file systems and very large file systems, enhanced system management with Red Hat Network, extended update support, and more.



Note

Please visit <http://www.redhat.com/rhel/add-ons/> for more information about available *Add-Ons for Red Hat Enterprise Linux*.

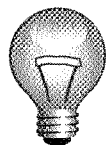
For information about other products which are provided by Red Hat, such as Red Hat Enterprise Virtualization, JBoss Enterprise Middleware, Red Hat Enterprise MRG, and various custom consulting and engineering services, <http://www.redhat.com/products/> also has useful information.

The Fedora Project also provides additional packages for Red Hat Enterprise Linux through *EPEL (Extra Packages for Enterprise Linux)*. EPEL is a volunteer-based community effort to create a repository of high-quality add-on packages which can be used with Red Hat Enterprise Linux and compatible derivatives. It accepts legally-unencumbered free and open source software which does not conflict with packages in Red Hat Enterprise Linux or Red Hat add-on products. EPEL packages are built for a particular major release of Red Hat Enterprise Linux and will be updated by EPEL for the standard support lifetime of that major release.

Red Hat does not provide commercial support or service level agreements for EPEL packages. While not supported officially by Red Hat, EPEL provides a useful way to reduce support costs for unsupported packages which your enterprise wishes to use with Red Hat Enterprise Linux. EPEL allows you to distribute support work you would need to do by yourself across other organizations which share your desire to use this open source software in RHEL. The software packages themselves go through the same review process as Fedora packages, meaning that experienced Linux developers have examined the packages for issues. As EPEL does not replace or conflict with software packages shipped in RHEL, you can use EPEL with confidence that it will not cause problems with your normal software packages.

For developers who wish to see their open source software become part of Red Hat Enterprise Linux, often a first stage is to sponsor it in EPEL so that RHEL users have the opportunity to use it, and so experience is gained with managing the package for a Red Hat distribution.

Visit <http://fedoraproject.org/wiki/EPEL/> for more information about EPEL.



Important

EPEL is supported by the community-managed Fedora Project and not by Red Hat Support.

Contacting Red Hat Technical Support

One of the benefits of your subscription to Red Hat Enterprise Linux is access to technical support through Red Hat's customer portal at <http://access.redhat.com/>. If you do not have a Red Hat account on the customer portal or are not able to log in, you can go to <https://access.redhat.com/support/faq/LoginAssistance.html> or contact Customer Service for assistance.

You may be able to resolve your problem without formal technical support by searching Knowledgebase (<https://access.redhat.com/kb/knowledgebase/>). Otherwise, Red Hat Support may be contacted through a web form or by phone depending on your support level. Phone numbers and business hours for different regions vary; see <https://access.redhat.com/support/contact/technicalSupport.html> for current information. Information about the support process is available at https://access.redhat.com/support/policy/support_process.html.

Some tips on preparing your bug report to most effectively engage Red Hat Support:

- *Define the problem.* Make certain that you can articulate the problem and its symptoms before you contact Red Hat. Be as specific as possible, and detail the steps you can use (if any) to reproduce the problem.
- *Gather background information.* What version of our software are you running? Are you using the latest update? What steps led to the failure? Can the problem be recreated and what steps are required? Have any recent changes been made that could have triggered the issue? Were messages or other diagnostic messages issued? What *exactly* were they (exact wording may be critical)?
- *Gather relevant diagnostic information.* Be ready to provide as much relevant information as possible; logs, core dumps, traces, the output of **sosreport**, etc. Technical Support can assist you in determining what is relevant.
- *Determine the Severity Level of your issue.* Red Hat uses a four-level scale to indicate the criticality of issues; criteria may be found at https://access.redhat.com/support/policy/GSS_severity.html.



Warning

Bugzilla is not a support tool! For support issues affecting Red Hat Enterprise Linux, customers should file their bugs through the support channels discussed above in order to ensure that Red Hat is fully aware of your issue and can respond under the terms of your Service Level Agreement. Customers should *not* file bugs directly in the <http://bugzilla.redhat.com/> web interface.

For Red Hat Enterprise Linux, Bugzilla is used by engineering to track issues and changes, and to communicate on a technical level with Engineering partners and other external parties. Anyone, even non-customers, can file issues against Bugzilla, and Red Hat does monitor them and review them for inclusion in errata.

However, Red Hat does not guarantee any SLA for bugs filed directly in Bugzilla (bypassing normal support channels). A review might happen immediately, or after a time span of any length. Issues coming through Support are always prioritized above issues of similar impact and severity filed against Bugzilla. Also, work arounds and hotfixes if possible and appropriate may be provided to customers by Support even before a permanent fix is issued through Red Hat Network.

Red Hat considers issues directly entered into Bugzilla important feedback, and it allows us to provide efficient interaction with the open source development community and as much transparency as possible to customers as issues are processed. Nevertheless, for customers encountering production issues in Red Hat Enterprise Linux, Bugzilla is not the right channel.

About This Course

RHCE Rapid Track Course

RHCE Rapid Track Course (RH300) provides a fast-track preparation for the Red Hat Certified Engineer (RHCE) exam for experienced, senior Linux system administrators already familiar with most of the topics covered in this class. This is a very fast paced course that combines the *RHCSA Rapid Track Course* (RH200) and *Red Hat System Administration III* (RH255), normally eight days of training, into a single four day course. Students move quickly through intermediate and advanced tasks, building upon their extensive existing knowledge of command-line based Linux system administration best practices.

Objectives

- Allow experienced senior Linux system administrators to review and fill in the gaps in their knowledge of system administration and their ability to configure and secure important network services on Red Hat Enterprise Linux
- Prepare highly experienced and motivated students to validate their skills in the RHCE exam

Audience and Prerequisites

- Students who are senior Linux system administrators with *at least* three years of full-time Linux experience, preferably using Red Hat Enterprise Linux
- Students should enter the class either with RHCT or RHCSA certification *or* have equivalent Linux skills

Structure of the Course

Red Hat training courses are interactive, hands-on, performance-based, real world classes meant to engage your mind and give you an opportunity to use real systems to develop real skills. We encourage students to participate in class and ask questions in order to get the most out of their training sessions.

This course is divided up into a number of *Units* organized around a particular topic area. Each Unit is divided up into multiple *Sections* which focus on a specific skill or task. The unit will start with an introduction to the material, then move on to the first section.

In each section, there will be a *presentation* led by the instructor. During the presentation, it may be a good idea to take notes in your student workbook (this book), and the instructor may remind you to do so. The presentation is followed by a short activity or *assessment* to give you the opportunity to practice with the material or review procedures. After a review of the assessment, the instructor will move on to the next section. At the end of the unit, there will normally be a hands-on lab exercise of some sort (a "*criterion test*") which will give you an opportunity to learn by doing and review your understanding of the unit's content. Please feel free ask questions in class, or asking the instructor for advice and help during the end-of-unit exercise. We want the

classroom environment to be a "low risk" place where you feel comfortable asking questions and learning from things that work and things that do not at first.

Orientation to the Classroom Network

Two subnets may be used in this course. The primary classroom network is 192.168.0.0/24, and belongs to hosts in the DNS domain "example.com". This network will be used for most classroom activities. Some courses use a second subnet, 192.168.1.0/24, belonging to hosts in the DNS domain "remote.test". This network can be reached from hosts in example.com, and is used in lab exercises which require testing services or security settings from machines (theoretically) outside your administrative control.

Students are each assigned a physical machine (desktopX.example.com on 192.168.0.X) which may host two or more virtual machines for lab activities, serverX.example.com and hostX.example.com.

In some courses, students may also use a non-root account on a test machine in the remote.test domain, remoteX.example.com (192.168.1.X) to test access to network services on their example.com machines in lab activities.

The instructor controls a number of machines which students may see as well. The machine instructor.example.com (also known as instructor.remote.test) is the classroom utility server, providing default routing services, DHCP, DNS name service, one or more YUM repositories of software used by the class, and other network services. It is also connected to the classroom video projector to allow the instructor to display slides and demonstrations. It provides a virtual machine for the instructor, demo.example.com, which the instructor will use for in-class demonstrations.

Machine name	IP addresses	Role
desktopX.example.com	192.168.0.X	Physical student workstation
serverX.example.com	192.168.0.(X+100)	Main student virtual machine
hostX.example.com	192.168.0.(X+200)	Secondary student virtual machine
remoteX.remote.test	192.168.1.X	Student test machine in remote.test domain (shared)
instructor.example.com	192.168.0.254	Physical instructor machine and utility server
instructor.remote.test	192.168.1.254	Identity of instructor.example.com on remote.test network
demo.example.com	192.168.0.250	Instructor virtual demonstration machine

Table1. Classroom Machines

Internationalization

Language Support

Red Hat Enterprise Linux 6 officially supports twenty-two languages: English, Assamese, Bengali, Chinese (Simplified), Chinese (Traditional), French, German, Gujarati, Hindi, Italian, Japanese, Kannada, Korean, Malayalam, Marathi, Oriya, Portuguese (Brazilian), Punjabi, Russian, Spanish, Tamil, and Telugu. Support for Maithili, Nepalese, and Sinhala are provided as Technology Previews.

System-wide Default Language

The operating system's default language is normally set to US English (en_US.UTF-8), but this can be changed during or after installation.

To use other languages, you may need to install additional package groups to provide the appropriate fonts, translations, dictionaries, and so forth. By convention, these package groups are always named **language-support**. These package groups can be selected during installation, or after installation with PackageKit (**System** → **Administration** → **Add/Remove Software**) or **yum**.

A system's default language can be changed with **system-config-language** (**System** → **Administration** → **Language**), which affects the `/etc/sysconfig/i18n` file.

Per-user Language Selection

Users may prefer to use a different language for their own desktop environment or interactive shells than is set as the system default. This is indicated to the system through the **LANG** environment variable.

This may be set automatically for the GNOME desktop environment by selecting a language from the graphical login screen by clicking on the **Language** item at the bottom left corner of the graphical login screen immediately prior to login. The user will be prompted about whether the language selected should be used just for this one login session or as a default for the user from now on. The setting is saved in the user's `~/.dmrc` file by GDM.

If a user wants to make their shell environment use the same **LANG** setting as their graphical environment even when they login through a text console or over **ssh**, they can set code similar to the following in their `~/.bashrc` file. This code will set their preferred language if one is saved in `~/.dmrc` or will use the system default if one is not:

```
i=$(grep 'Language=' ${HOME}/.dmrc | sed 's/Language=//')
if [ "$i" != "" ]; then
    export LANG=$i
fi
```

Languages with non-ASCII characters may have problems displaying in some environments. Kanji characters, for example, may not display as expected on a virtual console. Individual commands can be made to use another language by setting **LANG** on the command-line:

```
[user@host ~]$ LANG=fr_FR.UTF-8 date
lun. oct. 24 10:37:53 CDT 2011
```

Subsequent commands will revert to using the system's default language for output. The **locale** command can be used to check the current value of **LANG** and other related environment variables.

Input Methods

IBus (Intelligent Input Bus) can be used to input text in various languages under X if the appropriate language support packages are installed. You can enable IBus with the **im-chooser** command (**System** → **Preferences** → **Input Method**).

Language Codes Reference

Language	\$LANG value	Language package group
English (US)	en_US.UTF-8	(default)
Assamese	as_IN.UTF-8	assamese-support
Bengali	bn_IN.UTF-8	bengali-support
Chinese (Simplified)	zh_CN.UTF-8	chinese-support
Chinese (Traditional)	zh_TW.UTF-8	chinese-support
French	fr_FR.UTF-8	french-support
German	de_DE.UTF-8	german-support
Gujarati	gu_IN.UTF-8	gujarati-support
Hindi	hi_IN.UTF-8	hindi-support
Italian	it_IT.UTF-8	italian-support
Japanese	ja_JP.UTF-8	japanese-support
Kannada	kn_IN.UTF-8	kannada-support
Korean	ko_KR.UTF-8	korean-support
Malayalam	ml_IN.UTF-8	malayalam-support
Marathi	mr_IN.UTF-8	marathi-support
Oriya	or_IN.UTF-8	oriya-support
Portuguese (Brazilian)	pt_BR.UTF-8	brazilian-support
Punjabi	pa_IN.UTF-8	punjabi-support
Russian	ru_RU.UTF-8	russian-support

Language	\$LANG value	Language package group
Spanish	es_ES.UTF-8	spanish-support
Tamil	ta_IN.UTF-8	tamil-support
Telugu	te_IN.UTF-8	telugu-support
<i>Technology Previews</i>		
Maithili	mai_IN.UTF-8	maithili-support
Nepali	ne_NP.UTF-8	nepali-support
Sinhala	si_LK.UTF-8	sinhala-support

Table 2. Language Codes



UNIT ONE

SOFTWARE MANAGEMENT

Introduction

Topics covered in this unit:

- Registration of systems with Red Hat Network (RHN)
- Using yum to manage software packages
- Using rpm to get information about software packages
- Building your own RPM software packages
- Creating and using a yum package repository

Register with Red Hat Network (RHN)

What is Red Hat Network?

Red Hat Network is a centrally-managed service that makes it easy to deploy software and software updates to Red Hat Enterprise Linux systems and to remotely manage and monitor those systems. You can use the "hosted" RHN service managed by Red Hat, or you can set up and manage your own RHN Satellite in your organization. Either way, to get package updates for your clients from RHN and to have them show up in your web management interface, you need to start by registering those systems with the RHN server of your choice.

Using `rhn_register`

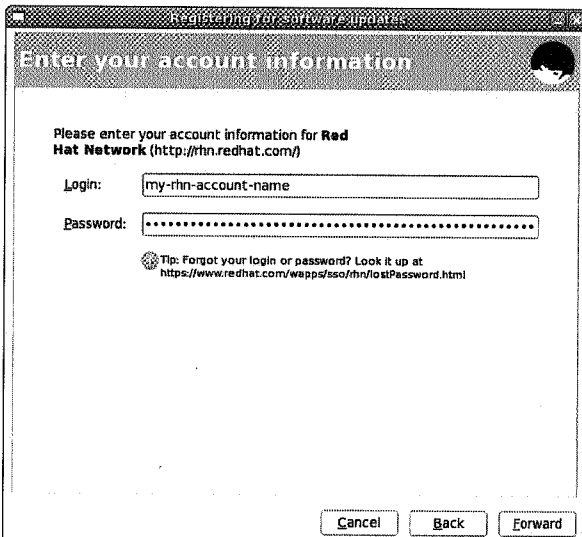
Start the Red Hat Network (RHN) registration process by running the **`rhn_register`** command from the command-line or choosing it from the GUI menu: **System** → **Administration** → **RHN Registration**

If you have a RHN Satellite or RHN Proxy server, choose the **I have access to a Red Hat Network Satellite...** button in the GUI. Fill in the DNS name of the RHN Satellite server or RHN Proxy server.

If you do not have a RHN Satellite or RHN Proxy server, or you want to register with Hosted RHN, choose the **I'd like to receive updates from Red Hat Network** button.

If you need to set proxy setting for the connection, click on the **Advanced Network Configuration...** button and fill in the appropriate fields.

Fill in your Red Hat Network account information. If you have forgotten your account name or password, or you need to create a new account, go to <https://www.redhat.com/wapps/sso/login.html>



The screenshot shows a window titled "Registering for software updates" with a sub-header "Enter your account information". The main text says "Please enter your account information for Red Hat Network (<http://rhn.redhat.com/>)". There are two input fields: "Login:" with the text "my-rhn-account-name" and "Password:" with a masked password. Below the password field is a tip: "Tip: Forgot your login or password? Look it up at <https://www.redhat.com/wapps/sso/rhn/lostPassword.html>". At the bottom are three buttons: "Cancel", "Back", and "Forward".

The next screen allows you to limit updates to maintain compatibility with Red Hat Enterprise Linux minor releases. If you want this ability choose **Limited updates**. If you want all the current updates, choose **All available updates**.

Select operating system release

Operating system version:

☐ **Limited updates** will be provided to this system to maintain compatibility with the following eligible Red Hat Enterprise Linux minor release software channel:

Minor release:

Tip: Minor releases with a '*' are currently fully supported by Red Hat.

☐ **All available updates** will be provided to this system. This system, if kept updated, will always be equivalent to the latest available minor release of Red Hat Enterprise Linux 5. It will be registered to the main 'Red Hat Enterprise Linux 5' software channel.

Warning: You will **not** be able to limit this system to a minor release that is older than the most recent minor release if you select this option.

Enter the name for your system (it will use the current hostname by default), and optionally send the hardware and package profile to RHN.

Create your system profile

System Name

You'll want to choose a name for this system so you'll be able to identify it in the Red Hat Network interface.

System Name:

Profile Data

You'll need to send us a profile of what packages and hardware are installed on your system so we can determine what updates are available.

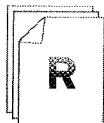
☒ Send hardware profile

☒ Send package profile



Note

The **rhn_register** command works equally well in a graphical environment or a text environment. If you run **rhn_register** in a text-only environment, it will prompt for information much as the GUI does.



References

`rhncregister(8)` and **`rhncplugin(8)`** man pages

Knowledgebase: "What is the command `rhncregister` used for in Red Hat Enterprise Linux?"
<https://access.redhat.com/kb/docs/DOC-11217>

Knowledgebase: "I had to re-install my system. How do I re-register my system with Red Hat Network (RHN)?"
<https://access.redhat.com/kb/docs/DOC-8037>

Red Hat Enterprise Virtualization for Servers 2.2: 5.5-2.2 Hypervisor Deployment Guide

- Section 5.1.7: Register to RHN



Practice Quiz

Red Hat Network Registration

1. The menu item that begins the registration with Red Hat Network is _____.
2. The first registration choice determines whether a system registers with _____ or _____.
3. Optionally additional _____ server and authentication information may need to be provided.
4. An _____ and its matching password must be provided for successful Red Hat Network registration.
5. The last questions to be answered during the registration process are _____ and whether to upload _____ and _____ profile information.

Using Third-Party Repositories

Third-party repositories are network-accessible directories of software package files which can be accessed by **yum**, provided outside of Red Hat Network. Yum repositories are used by non-Red Hat distributors of software, or for small collections of local packages. (For example, Adobe provides some of its free software for Linux through a yum repository.) The **instructor** classroom server actually hosts yum repositories for this class.

Put a file in the `/etc/yum.repos.d/` directory to enable support for a new third-party repository. Repository configuration files must end in `*.repo`. The repository definition contains the URL of the repository, a name, whether to use GPG to check the package signatures, and if so the local file containing the trusted GPG key.

Examples of `/etc/yum.repos.d/*.repo` configuration files:

An example with a single repository, with security checks of downloaded packages disabled:

```
[GLS]
name=Instructor GLS Repository
baseurl=ftp://instructor.example.com/pub/gls
gpgcheck=0
```

An example with multiple repository references in a single file:

```
[base]
name=Instructor Server Repository
baseurl=http://instructor.example.com/pub/rhel6/dvd
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# Optional rhel6
[optional]
name=Instructor Optional Repository
baseurl=http://instructor.example.com/pub/rhel6/Optional
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[client]
name=Instructor Client Repository
baseurl=http://instructor.example.com/pub/rhel6/Client
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
enabled=0

[kernel-extras]
name=Instructor Kernel Extras Repository
baseurl=http://instructor.example.com/pub/rhel6/Kernel-Extras
gpgcheck=1
```




Note

Note that some repositories, such as EPEL (Extra Packages for Enterprise Linux), provide this configuration file as part of an RPM package that can be downloaded from the web, and installed with **yum localinstall**.

Installing the Red Hat Enterprise Linux 6 EPEL repo package:

```
[root@serverX ~]# rpm --import http://download.fedora.redhat.com/pub/epel/RPM-GPG-KEY-EPEL-6
[root@serverX ~]# yum install http://download.fedora.redhat.com/pub/epel/beta/6/x86_64/epel-release-6-5.noarch.rpm
[root@serverX ~]# cat /etc/yum.repos.d/epel.repo
```

```
[epel]
name=Extra Packages for Enterprise Linux 6 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6

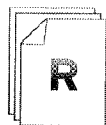
[epel-debuginfo]
name=Extra Packages for Enterprise Linux 6 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch/debug
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 6 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/6/SRPMS
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-6&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
gpgcheck=1
```



Important

Install the RPM GPG key before installing signed packages. This will verify that the package belongs to a key you have imported. Otherwise, **yum** will complain about the missing key. (You can use the **--nogpgcheck** option to ignore missing GPG keys, but this could cause forged or insecure packages to be installed on your system.)



References

yum(1) and **yum.conf**(5) man pages



Practice Exercise

Using YUM repositories

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You will configure your server to use a separate YUM repository to obtain updates, and update your machine.

1. Create the file **/etc/yum.repos.d/errata.repo**, to enable the "Updates" repository found on the instructor machine. It should access content found at the following URL: `ftp://instructor.example.com/pub/rhel6/Errata`
2. Update all relevant software provided by the repository, using **yum update**.

Using yum

yum is a powerful command-line tool that can be used to more flexibly manage software packages.

PackageKit uses **yum** to get packages. Official Red Hat packages are normally downloaded from Red Hat Network (RHN). When you register your machine with RHN, **yum** is automatically configured to use it. You can also configure **yum** to get packages from third-party package repositories over the network.

Basic yum Commands

1. **yum help** will display usage information
2. **yum list** displays installed and available packages
3. **yum search KEYWORD** lists packages by keywords
4. **yum info PACKAGENAME** gives detailed information about a package
5. **yum install PACKAGENAME** obtains and installs a software package, including any dependencies
6. **yum remove PACKAGENAME** removes an installed software package, including any supported packages
7. **yum update PACKAGENAME** obtains and installs a newer version of the software package, including any dependencies. Generally the process tries to preserve configuration files in place, but in some cases they may be renamed if the packager thinks the old one will not work after update. With no **PACKAGENAME** specified, it will install all relevant updates.

Use this space for notes

Example yum commands:

To search for packages that have "web server" in their description, summary, or package name:

```
[root@serverX ~]# yum search 'web server'
===== Matched: web server =====
mod_auth_mysql.x86_64 : Basic authentication for the Apache web server using a
                        : MySQL database
webalizer.x86_64 : A flexible Web server log file analysis program
freeradius.x86_64 : High-performance and highly configurable free RADIUS server
hsqldb.x86_64 : Hsqldb Database Engine
htdig.x86_64 : ht://Dig - Web search engine
htdig-web.x86_64 : Scripts and HTML code needed for using ht://Dig as a web
                  : search engine
httpd.x86_64 : Apache HTTP Server
...
```

To get information on the Apache HTTP Server:

```
[root@serverX ~]# yum info httpd
Available Packages
Name       : httpd
Arch       : x86_64
Version    : 2.2.15
Release    : 5.el6
Size       : 811 k
Repo       : base
Summary    : Apache HTTP Server
URL        : http://httpd.apache.org/
License    : ASL 2.0
Description: The Apache HTTP Server is a powerful, efficient, and extensible
           : web server.
```

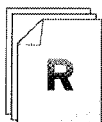
To install, update and remove the **httpd** package:

```
[root@serverX ~]# yum install httpd
[root@serverX ~]# yum update httpd
[root@serverX ~]# yum remove httpd
```



Warning

yum remove will remove the package(s) listed *and any package that requires the package(s) being removed* (and package(s) which require those packages, and so on). This can lead to unexpected removal of packages, so carefully check the list of packages to be removed.



References

yum(1), **yum.conf(5)** man pages



Practice Exercise

Searching for and installing packages

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Login as **root** on serverX and perform the following tasks:

1. Attempt to run the command **gnuplot**. You should find that it is not installed.
2. Search for plotting packages.
3. Find out more information about the **gnuplot** package.
4. Install the **gnuplot** package.
5. Attempt to remove the **gnuplot** package, but say no.
How many packages would be removed? _____
6. Attempt to remove the **gnuplot-common** package, but say no.
How many packages would be removed? _____

Handling Third-Party Software

The **rpm** utility is a low-level tool that is useful to get information about the contents of package files and installed packages. **rpm** can be used to get detailed information about packages from package files or the local database of information about installed packages.

RPM queries: Information about versions of packages

- **-q -a** - all installed packages
- **-q *PACKAGENAME*** - currently installed *PACKAGENAME*
- **-q -p *PACKAGEFILE.rpm*** - package file named *PACKAGEFILE.rpm*
- **-q -f *FILENAME*** - what package provides *FILENAME*

RPM queries: Information about content of packages

- **-q** - lists the package's name and version; compare to **yum list**
- **-q -i** - package information; compare to **yum info**
- **-q -l** - list of files installed by the specified package
- **-q --configfiles** - list just the configuration files
- **-q --docfiles** - list just the documentation files
- **-q --scripts** - list shell scripts that may run after the package is installed or uninstalled



Note

The **repoquery** command can also be used to get information about packages and their contents. It differs from **rpm** by looking up that information in yum's repositories and RHN instead of the local database of installed packages.

Using yum to install local package files

yum localinstall *PACKAGEFILE.rpm* can be used to install package files directly. It automatically downloads any dependencies the package has from RHN and any configured **yum** repositories. Packages are normally digitally signed to ensure they are legitimate; if the package is not signed by a key trusted by your system, it will be rejected. The **--nogpgcheck** option can disable the signature check if you are certain the package is legitimate.



Note

rpm -ivh *PACKAGEFILE.rpm* can also be used to install package files. However, using **yum** helps maintain a transaction history kept by **yum** (see **yum history**).

Example rpm query commands:

Querying installed packages:

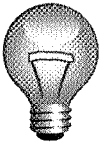
```
[root@serverX ~]# rpm -q samba-client
samba-client-3.5.4-68.el6.x86_64
[root@serverX ~]# rpm -ql zlib
/lib64/libz.so.1
/lib64/libz.so.1.2.3
/usr/share/doc/zlib-1.2.3
/usr/share/doc/zlib-1.2.3/ChangeLog
/usr/share/doc/zlib-1.2.3/FAQ
/usr/share/doc/zlib-1.2.3/README
[root@serverX ~]# rpm -q --scripts httpd
preinstall scriptlet (using /bin/sh):
# Add the "apache" user
getent group apache >/dev/null || groupadd -g 48 -r apache
getent passwd apache >/dev/null || \
    useradd -r -u 48 -g apache -s /sbin/nologin \
        -d /var/www -c "Apache" apache
exit 0
postinstall scriptlet (using /bin/sh):
# Register the httpd service
/sbin/chkconfig --add httpd
preuninstall scriptlet (using /bin/sh):
if [ $1 = 0 ]; then
    /sbin/service httpd stop > /dev/null 2>&1
    /sbin/chkconfig --del httpd
fi
posttrans scriptlet (using /bin/sh):
/sbin/service httpd condrestart >/dev/null 2>&1 || :
```

Querying and installing package files:

```
[root@serverX ~]# cd /net/instructor/var/ftp/pub/materials/
[root@serverX ~]# rpm -qpl wonderwidgets-1.0-4.x86_64.rpm
/etc/wonderwidgets.conf
/usr/bin/wonderwidgets
/usr/share/doc/wonderwidgets-1.0
/usr/share/doc/wonderwidgets-1.0/README.txt
[root@serverX ~]# rpm -qpi wonderwidgets-1.0-4.x86_64.rpm
Name       : wonderwidgets                Relocations: (not relocatable)
Version    : 1.0                        Vendor: Red Hat, Inc.
Release    : 4                          Build Date: Fri 03 Dec 2010 05:42:55 AM EST
Install Date: (not installed)           Build Host: station166.rosemont.lan
Group      : GLS/Applications            Source RPM: wonderwidgets-1.0-4.src.rpm
Size       : 4849                        License: GPL
Signature  : (none)
Summary    : Demonstration package for use in GLS training.
Description:
A demonstration package that installs an executable, and a config file.
```

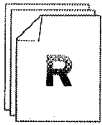


```
[root@serverX ~]# rpm -qp --configfiles wonderwidgets-1.0-4.x86_64.rpm
/etc/wonderwidgets.conf
[root@serverX ~]# rpm -qp --docfiles wonderwidgets-1.0-4.x86_64.rpm
/usr/share/doc/wonderwidgets-1.0/README.txt
[root@serverX ~]# yum localinstall wonderwidgets-1.0-4.x86_64.rpm
...
Package wonderwidgets-1.0-4.x86_64.rpm is not signed
[root@serverX ~]# yum localinstall --nogpgcheck wonderwidgets-1.0-4.x86_64.rpm
[root@serverX ~]# rpm -q wonderwidgets
wonderwidgets-1.0-4.x86_64
```



Important

Be careful when installing packages from third parties, not just because of the software that they may install, but because the RPM may run arbitrary scripts as **root** as part of the installation process.



References

rpm(8) and **repoquery(1)** man pages



Practice Exercise

Handling Third-Party Software

Carefully perform the following steps. Ask your instructor if you have problems or questions.

In this exercise you will gather information about a third-party package, extract files from it, and install it as a whole on your desktopX system.

1. Download *wonderwidgets-1.0-4.x86_64.rpm* from <http://instructor/pub/materials>.
2. What files does it contain?
3. What scripts does it contain?
4. How much disk space will it use when installed?
5. Use **yum localinstall** to install the package.

RPM Package Design

Managing software in the form of RPM packages is much simpler than working with software which has simply been extracted into a file system from an archive. It lets you track which files were installed by the software package, which ones need to be removed if it is uninstalled, and check to ensure supporting packages are present when it is installed.

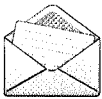
Therefore, it is useful to know how to create RPM packages for your own software. For the remainder of this unit, we will look at how to create a basic RPM package and point you to resources which will help you learn how to create more complex packages as your skills grow.

Design/Structure of an RPM

Each RPM package is made up of three basic components:

- **metadata** - Data about the package: the package name, version, release, builder, date, dependencies, etc.
- **files** - archive of files provided by the package (including file attributes)
- **scripts** - these execute when the package is installed, updated, and/or removed

When building an RPM package, the metadata about the package needs to be specified, the files in the archive need to be provided, and the scripts that should be run when the package is installed or uninstalled need to be embedded.

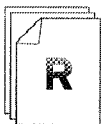


Note

Internally, files are stored as a **cpio** archive inside the package file. The **rpm2cpio** command can be used to extract them to the current working directory without installing the package: **rpm2cpio package-1.2.3-4.el6.x86_64.rpm | cpio -id**

The following **rpm** queries are useful for investigating the structure of an RPM package:

- **rpm -qd** - list documentation files (%doc)
- **rpm -qc** - list configuration files (%config)
- **rpm -q --scripts** - list %pre, %post, %preun, and %postun scripts



References

Red Hat Enterprise Linux Deployment Guide, Section 3.2.6: Querying RPM

rpm(8), **rpm2cpio(8)**, and **cpio(1)** man pages

RPM Package Specifications

To construct an RPM package, you will need a build specification file or *spec file*. A spec file is simply a text file that contains information on how to build the installable RPM package. You can think of it as being roughly divided into five parts:

- The *introduction* or *preamble*, listing metadata about the package (name, version, license, etc.)
- The *build* instructions, which specify how to compile and prepare the software
- The *scriptlets*, which specify commands to run on install, uninstall, or upgrade
- The *manifest*, a list of files to package and their permissions on package installation
- The *changelog*, which tracks changes made to this RPM package

Important preamble directives:

- **Name** - The name of the package, usually chosen by the developers. For detailed guidance, look at the Fedora Naming Guidelines, at <http://fedoraproject.org/wiki/Packaging:NamingGuidelines>
- **Version** - The version of the package (usually numeric), usually chosen by the developers.
- **Release** - The release of the package, chosen by the packager. This should increase each time you release a new package for distribution if you still use the same Version of the software.
- **Group** - The group to which the package belongs. See `/usr/share/doc/rpm-*/GROUPS` for the default set of groups, or use one of your own. This field is semi-obsolete, and is not related to **yum** package groups.
- **License** - The "Short License identifier" of the license used for the software. Detailed guidance on how to set this in a standard way can be found at <http://fedoraproject.org/wiki/Packaging/LicensingGuidelines>
- **Summary** - A short one-line description of the software. (Keep to about 50 characters or less.)
- **Source** - The file to be used as the source code. If there are more than one file used as source, add a number. E.g., `Source0`, `Source1`, `Source2`, etc.
- **BuildArch** - The architecture to use when building the package. Defaults to the system architecture. A common argument is **noarch**, which means that the package is architecture independent (often these packages consist of scripts or data files).
- **Requires** - A list of explicit requirements this package depends on. This could be a list of files or other packages. **rpmbuild** can generally autodetect most library dependencies, but there are some cases where you may need to list an explicit dependency. See <http://fedoraproject.org/wiki/Packaging/Guidelines#Requires> for additional guidance on **Requires**.
- **BuildRequires** - A list of requirements that are needed to *build* this package. This is a list with similar syntax to that of **Requires**, for example **BuildRequires: /usr/bin/gcc, gimp-libs >= 2.6.11**. See the link in **Requires** above for information on how to tell if you need missing **BuildRequires**.

Required spec file sections:

- **%description** section - A long description of the software. No line should be more than 80 characters long, but you may have multiple lines.
- **%prep** section -
- **%build** section -
- **%install** section -
- **%clean** section -
- **%files** section -
- **%changelog** section -

rpmbuild Steps

When **rpmbuild** runs, the build process will work through the following sections in order:

1. **%prep**
2. **%build**
3. **%install**
4. Package the completed RPM
5. **%clean**

Creating a new spec file

On Red Hat Enterprise Linux 6, **vim** has a macro that helps to create a specification file. Simply pass a file name that ends in **.spec**:

```
[student@serverX]$ vim foo.spec
```

vim will use the spec template to provide some common entries for RPM building.



Note

When an RPM package is built, a *source RPM* (SRPM) package is also created, with an architecture of **src**. Another way to get a spec file is to install a source package by running **rpm -ivh package-1.2.3-4.src.rpm** as a *non-root user*. The spec file for the package will be in **~/rpmbuild/SPECS**.

Example spec file

An annotated example of a spec file follows.

```
%define debug_package %{nil} ❶
%define product_family Red Hat Enterprise Linux
%define release_name Santiago
%define base_release_version 6
%define full_release_version 6.0
%define beta Beta

Name:          redhat-release ❷
Version:       %{base_release_version} ❸
Release:       6.0.0.24%{?dist} ❹
Summary:       %{product_family} release file ❺
Group:         System Environment/Base
License:       GPLv2
Obsoletes:     rawhide-release redhat-release-as redhat-release-es redhat-release-ws ❻
Source0:       redhat-release-6-4.tar.gz ❼

%description ❸
%{product_family} release files

%prep ❹
%setup -q

%build ❺
echo OK

%install ❻
rm -rf $RPM_BUILD_ROOT

# create /etc
mkdir -p $RPM_BUILD_ROOT/etc

# create /etc/system-release and /etc/redhat/release
echo "%{product_family} release %{full_release_version}%{?beta: %{beta}}
(%{release_name})" > $RPM_BUILD_ROOT/etc/redhat-release
ln -s redhat-release $RPM_BUILD_ROOT/etc/system-release

# write cpe to /etc/system-release-cpe
echo "cpe:/o:redhat:enterprise_linux:%{version}:%{?beta:%{beta}}%{!?beta:GA}" >
  $RPM_BUILD_ROOT/etc/system-release-cpe

# create /etc/issue and /etc/issue.net
cp $RPM_BUILD_ROOT/etc/redhat-release $RPM_BUILD_ROOT/etc/issue
echo "Kernel \r on an \m" >> $RPM_BUILD_ROOT/etc/issue
cp $RPM_BUILD_ROOT/etc/issue $RPM_BUILD_ROOT/etc/issue.net
echo >> $RPM_BUILD_ROOT/etc/issue

# copy yum repos to /etc/yum.repos.d
mkdir -p $RPM_BUILD_ROOT/etc/yum.repos.d
for file in *.repo; do
    install -m 644 $file $RPM_BUILD_ROOT/etc/yum.repos.d
done

# copy GPG keys
mkdir -p -m 755 $RPM_BUILD_ROOT/etc/pki/rpm-gpg
for file in RPM-GPG-KEY* ; do
    install -m 644 $file $RPM_BUILD_ROOT/etc/pki/rpm-gpg
done
```

```

# set up the dist tag macros
install -d -m 755 $RPM_BUILD_ROOT/etc/rpm
cat >> $RPM_BUILD_ROOT/etc/rpm/macros.dist << EOF
# dist macros.

%%rhel %{base_release_version}
%%dist .el%{base_release_version}
%%el%{base_release_version} 1
EOF

%clean 12
rm -rf $RPM_BUILD_ROOT

%files 13
%defattr(-,root,root)
%doc EULA GPL autorun-template
%attr(0644,root,root) /etc/redhat-release
/etc/system-release
%config %attr(0644,root,root) /etc/system-release-cpe
%config(noreplace) %attr(0644,root,root) /etc/issue
%config(noreplace) %attr(0644,root,root) /etc/issue.net
%config %attr(0644,root,root) /etc/yum.repos.d/*
%dir /etc/pki/rpm-gpg
/etc/pki/rpm-gpg/*
/etc/rpm/macros.dist

%changelog 14
* Mon Mar 29 2010 Dennis Gregorovic <dgregor@redhat.com> - 6-6.0.0.24
- Add beta debuginfo repos
- Resolves: rhbz#572308

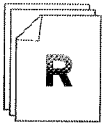
```

- ❶ Macros (like variables) that can be used in the spec file
- ❷ The name of the package
- ❸ The version of the package. Notice it uses the **%{base_release_version}** macro defined above.
- ❹ The release of the package
- ❺ A short summary
- ❻ A list of package names that this package makes obsolete. If you had one of these packages installed on your machine, an update of this package would remove that package.
- ❼ A source file
- ❽ A long description
- ❾ The **%prep** section. Unfortunately, the RPM spec file uses % for sections as well as macros. **%prep** is a section, **%setup** is a macro.
- ❿ The **%build** section
- ⓫ The **%install** section. **\$RPM_BUILD_ROOT** is a variable that expands to the "build root". Files are copied from the build directory to **\$RPM_BUILD_ROOT**, as if **\$RPM_BUILD_ROOT** was / on the file system the software will be installed in. Then the contents of **\$RPM_BUILD_ROOT** listed in **%files** will be packaged into the final RPM file. You must create all necessary directories in **\$RPM_BUILD_ROOT** before copying files to that location. Source files can be referenced using a relative path from the top-level **%{name}-%{version}** source directory. For example, if you wanted to have a file placed in **/root/bin/** (found in the **%{name}-%{version}/bin** directory), you would need to do something like the following:

```
mkdir -p $RPM_BUILD_ROOT/root/bin
cp bin/my-script $RPM_BUILD_ROOT/root/bin
```

- 12 The **%clean** section. Normally clean only has the **rm** command above.
- 13 The list of files to be included in this package. Note that **%defattr** sets the default permissions the files will have, **%attr** can override that on a file-by-file basis. **%config** and **%doc** mark configuration files and documentation respectively. **%dir** marks a directory owned by the package. See http://fedoraproject.org/wiki/Packaging:Guidelines#File_and_Directory_Ownership and http://fedoraproject.org/wiki/Packaging:Guidelines#Configuration_files for more information.
- 14 The **%changelog** section is for the packager to list items that changed in this release. Newest entries to the changelog go at the start of the section. Each entry has the format seen in the example, and entries are separated by a blank line.

The example above does not use any scriptlets. For more information on scriptlets, see the draft Fedora RPM Guide referenced below.



References

Fedora RPM Guide -

http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/

Fedora Packaging Guidelines -

<http://fedoraproject.org/wiki/Packaging:Guidelines>



Practice Quiz

RPM Spec File

1. The package _____ is usually derived from the open source project while the package _____ is the packager's version.
2. The _____ directive categorizes the type of package being built.
3. The name of the tarball containing the files used to build the package is specified with the _____ directive.
4. The _____ directive specifies the target architecture the package is being built for. _____ will be its value when the package can be installed on any architecture.
5. The _____ directive specifies the 1-line description of a package while the _____ section provides a more thorough explanation of what that package is for.
6. The _____ section contains the code used to place files in the _____ chroot directory structure.
7. The _____ section defines which files and directories to package into the RPM.
8. The _____, _____, and _____ sections contain shell code used to assemble a package and clean up after it has been built.

Building and Signing an RPM Package

The five steps for building an RPM package:

1. *Tarball*

Get the tar file containing the source. By default **rpmbuild** assumes the top-level directory of the archive is named **%{name}-%{version}**. Place this file in the **~/rpmbuild/SOURCES/** directory.

2. *Spec file*

Create a spec file and populate the required fields. Place this file in the **~/rpmbuild/SPECS/** directory.

3. *rpmbuild*

Use the **rpmbuild** command to build the package(s). For example,

```
rpmbuild -ba demo.spec
```

4. *Sign*

Use a GPG key to sign the RPM package. You can use **rpmbuild -ba --sign demo.spec** to build and sign the package in one step. If the package is already built, use **rpm --resign demo-1.0-1.x86_64.rpm** to add (or change) a GPG signature.

5. *Test*

Test the package by installing it on a development system to ensure the correct payload, scripts, etc.

Preparing a GPG Signing Key

RPM packages are normally digitally signed so that users can verify that a package actually came from the preparer it claims to belong to. This helps to block forged packages from being installed if a yum repository is compromised in some way. The next few steps detail how to create your own signing key. Once you have a signing key you can use it for signing many packages.

If you do not have a GPG key yet, run the **gpg --gen-key** command to generate a new one.



Note

You must have a graphical session open to run **gpg --gen-key**. It uses a graphical box to accept your input for the passphrase.

```
[student@serverX ~]$ gpg --gen-key
gpg (GnuPG) 2.0.14; Copyright (C) 2009 Free Software Foundation, Inc. This is free
software: you are free to change and redistribute it. There is NO WARRANTY, to the extent
```

```

permitted by law. Please select what kind of key you want: (1) RSA and RSA (default) (2)
DSA and Elgamal (3) DSA (sign only) (4) RSA (sign only) Your selection? Enter
RSA keys may be between 1024 and 4096 bits long. What keysize do you want? (2048) Enter
Requested keysize is 2048 bits Please specify how long the key should be valid. 0 = key
does not expire <n> = key expires in n days <n>w = key expires in n weeks <n>m = key
expires in n months <n>y = key expires in n years Key is valid for? (0) Enter
Key does not expire at all Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key. Real name: My Name
Email address: student@serverX.example.com
Comment: Enter
You selected this USER-ID: "My Name <student@serverX.example.com>" Change (N)ame,
(C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key. Enter passphrase Passphrase: testing123
Please re-enter this passphrase. Passphrase: testing123
We need to generate a lot of random bytes. It is a good idea to perform some other action
(type on the keyboard, move the mouse, utilize the disks) during the prime generation;
this gives the random number generator a better chance to gain enough entropy.

```

```

gpg: /home/student/.gnupg/trustdb.gpg: trustdb created
gpg: key 54AF5285 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/54AF5285 2010-12-09
    Key fingerprint = 315F E90B 1745 2288 EBAE 4E7B 4BC6 4568 54AF 5285
uid   My Name <student@serverX.example.com>
sub   2048R/D08B2951 2010-12-09

```

Find the public key ID from the output of **gpg --gen-key**, or run **gpg --fingerprint**

```

[student@serverX ~]$ gpg --fingerprint
/home/student/.gnupg/pubring.gpg
-----
pub 2048R/54AF5285 2010-12-09
    Key fingerprint = 315F E90B 1745 2288 EBAE 4E7B 4BC6 4568 54AF 5285
uid   My Name <student@serverX.example.com>
sub   2048R/D08B2951 2010-12-09

```

The public key ID is the string of eight hexadecimal characters after **pub 2048R/** (54AF5285 in the example above).

Export the public key (make sure you use your own key ID):

```

[student@serverX ~]$ gpg -a -o ~/RPM-GPG-KEY-student --export 54AF5285

```

Add the following to the **~/.rpmmacros** file (replacing the eight-character key ID with the key ID for your system) so RPM will sign packages with the key you created above.

```

[student@serverX ~]$ echo '%_gpg_name 54AF5285' > ~/.rpmmacros

```

Example RPM Package Build

The following shows an example of building an RPM package. The name of the package is **test**, version is **1.0** and release is **1**. It will provide a single file, **/usr/local/bin/myscript**, which simply runs the **date** command.

Create the directory, file and tarball:

```
[student@serverX ~]$ mkdir test-1.0
[student@serverX ~]$ cat << EOF > test-1.0/myscript
#!/bin/bash
date
EOF
[student@serverX ~]$ tar czvf test-1.0.tar.gz test-1.0
```

Create a spec file using **vim** in your home directory:

```
[student@serverX ~]$ vim test.spec
```



Note

In Red Hat Enterprise Linux 6, **vim** will automatically create a template spec file when you open a new file with a name that ends in **.spec**.

Fill in the fields as follows.

```
Name:          test
Version:       1.0
Release:       1%{?dist}
Summary:       A test package

Group:         Testing
License:       GPL
URL:           http://www.example.com/testing
Source0:       %{name}-%{version}.tar.gz1
BuildRoot:     %(mktemp -ud %{{_tmppath}}/%{name}-%{version}-%{release}-XXXXXX)

BuildRequires: /bin/rm, /bin/mkdir, /bin/cp2
Requires:      /bin/bash, /bin/date

%description
A testing package meant to deploy a single file.

%prep
%setup -q

%build
#configure3
#make %{{?_smp_mflags}}

%install
```

```
rm -rf $RPM_BUILD_ROOT
#make install DESTDIR=$RPM_BUILD_ROOT
mkdir -p $RPM_BUILD_ROOT/usr/local/bin
cp myscript $RPM_BUILD_ROOT/usr/local/bin

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root,-)
#%doc

%attr(0755,root,root)/usr/local/bin/myscript④

%changelog
* Thu Dec 09 2010 Forrest <forrest@redhat.com> 1.0-1
- Initial RPM
- Added /usr/local/bin/myscript
```

- ① The `%{name}` and `%{version}` macros are defined from the **Name:** and **Version:** lines above. Alternately, you could have used `test-1.0.tar.gz`.
- ② `rm`, `mkdir` and `cp` all come from the **coreutils** package, so you could have specified that package instead of the commands. These are the commands that are used in the **%install** section.
- ③ There are some macros that run even if they are commented, and **%configure** is one of them. If you comment **%configure** like: `#%configure`, it will complain about not finding `./configure`. Remove the **%configure** line entirely or remove the `%` from **configure**.
- ④ The `%attr` was added to force the permission to 0755. Notice that the `%defattr` has a `-` in the permissions place. This means that the files will get the same permissions that they have inside the tarball. An alternate means to produce the same result would be to run `chmod 755 test-1.0/myscript` and rebuild the tarball.

Install the **rpm-build** package as root:

```
[root@serverX ~]# yum install -y rpm-build
```

Run **rpmbuild** as student. The first time you run it, you will get an error. You will fix the error shortly. Running the **rpmbuild** command will create the directory structure needed to build the RPM package.

```
[student@serverX ~]$ rpmbuild test.spec
error: File /home/student/rpmbuild/SOURCES/test-1.0.tar.gz: No such file or directory
```



Warning

You should always run **rpmbuild** to build packages as a *non-root* user. *Do not build packages as root*. The reason for this is that mistakes in the spec file, especially in the **%install** and **%clean** sections, are more likely to damage your build machine's installation if run as root.

Copy the files to the correct location:

```
[student@serverX ~]$ cp test-1.0.tar.gz rpmbuild/SOURCES/
[student@serverX ~]$ cp test.spec rpmbuild/SPECS/
[student@serverX ~]$ cd rpmbuild/SPECS/
```

Build and sign the package:

```
[student@serverX ~]$ rpmbuild --sign -ba test.spec
Enter pass phrase: testing123
Pass phrase is good.
...
```

Look for errors in the output of **rpmbuild** and fix any issues you find. If there are no errors, you should find:

```
...
Wrote: /home/student/rpmbuild/SRPMS/test-1.0-1.el6.src.rpm
Wrote: /home/student/rpmbuild/RPMS/x86_64/test-1.0-1.el6.x86_64.rpm
...
```

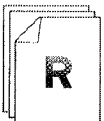
Test the package by installing the key, installing the package and running the command:

```
[root@serverX ~]# rpm --import /home/student/RPM-GPG-KEY-student
[root@serverX ~]# cd /home/student/rpmbuild/RPMS/x86_64
[root@serverX ~]# yum localinstall test-1.0-1.el6.x86_64.rpm
[student@serverX ~]$ /usr/local/bin/myscript
Thu Dec 09 10:21:53 EST 2010
```



Note

When reviewing a completed package for release, you may find the formal Fedora Package Review Guidelines (in the References below) to be useful.



References

Fedora RPM Guide -

http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/

Fedora Packaging Guidelines -

<http://fedoraproject.org/wiki/Packaging:Guidelines>

Fedora Package Review Guidelines -

<http://fedoraproject.org/wiki/Packaging:ReviewGuidelines>

rpmbuild(8) man page

Publish RPM Packages

Once you have an RPM package, you need to have a way to distribute it to your Red Hat Enterprise Linux systems. Ideally, you have a Red Hat Network Satellite server and can use that to deploy and manage your custom packages. But if you do not, one easy way to make packages available to clients is to set up your own yum repository.

Create a yum repository

```
[root@serverX ~]# yum install -y createrepo
[root@serverX ~]# mkdir -p /var/www/html/repo/Packages
[root@serverX ~]# cp test-1.0-1.el6.x86_64.rpm /var/www/html/repo/Packages
[root@serverX ~]# createrepo -v /var/www/html/repo/
[root@serverX ~]# cp /home/student/RPM-GPG-KEY-student /var/www/html/repo/
```

Sample yum configuration file

```
[example]
name=example
description=Example Yum Repository
baseurl=http://serverX.example.com/repo
enabled=1
gpgcheck=1
gpgkey=http://serverX.example.com/repo/RPM-GPG-KEY-student
```

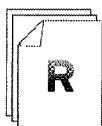
The **gpgkey** line could also look like the following, referencing an FTP server or a local file:

```
gpgkey=ftp://serverX/pub/RPM-GPG-KEY-student
```

-OR-

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-student
```

Whichever you choose, the GPG key file referenced is the public key matching the private GPG key that was used to sign the packages in the repository.



References

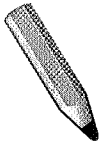
createrepo(8) man page



Practice Quiz

Create a Yum Repository Quiz

1. Install the _____ package if necessary.
2. Create a directory that can be
_____.
3. Create a subdirectory called _____.
4. Copy _____ to be
published into _____.
5. Execute _____ on the
_____ directory.



Test

Criterion Test

Performance Checklist

Create an RPM

- ☐ Download the file `ftp://instructor.example.com/pub/materials/hello.sh`.
- ☐ Create a simple RPM that installs **hello.sh** in **/root/bin**. Make sure that **hello.sh** is installed with a mode of 755.
- ☐ Create a GPG key and sign the package with the key. Export the public GPG key.



Note

You must have a graphical session available to successfully generate a GPG key. **gpg** now uses a graphical application to enter and validate the key.

- ☐ Deploy a web server and create a yum repository in **/var/www/html/Packages/**. Create a repository file that references `http://serverX/Packages`. Serve the GPG key from the web server and include the key in the repository file.
- ☐ Install your rpm using the yum repository above and run **/root/bin/hello.sh**.



Personal Notes



Unit Summary

Register with Red Hat Network (RHN)

In this section you learned how to:

- Register a system with Red Hat Network

Using Third-Party Repositories

In this section you learned how to:

- Manage repository definition files in **/etc/yum.repos.d/**

Using yum

In this section you learned how to:

- List packages by name, keyword, or file
- Get the version and description of a package
- Install, update and remove packages with **yum**

Handling Third-Party Software

In this section you learned how to:

- Query third-party packages for files before installation
- Query rpm package internals

RPM Package Design

In this section you learned how to:

- Use **rpm** to explore the structure of a package file

RPM Package Specifications

In this section you learned how to:

- Write a "spec file" to build your own RPM software package

Building and Signing an RPM Package

In this section you learned how to:

- Use **rpmbuild** to build and sign a new RPM package file

Publish RPM Packages

In this section you learned how to:

- Create your own **yum** repository to deploy a small number of package files



UNIT TWO

NETWORK MANAGEMENT

Introduction

Topics covered in this unit:

- Using command-line tools to view network settings
- Changing network settings in configuration files
- Troubleshooting networking problems
- Configuring multiple IP addresses on a single NIC
- Ethernet bonding two network interface cards together
- Basic tuning of networking-related kernel parameters

Understanding Network Configuration Files

As the instructor demonstrates the command or configuration file (or from the notes following), fill in this summary below of how to view and where to change the network configuration.

Setting Category	View Current Configuration	Change Configuration
IP Address and Subnet Mask		
Routing/Default Gateway		
System Hostname		
Name Resolution		

Table 2.1. Network Configuration from the Command-Line

Network Interface Names

The Linux kernel names interfaces with a specific prefix depending on the type of interface. For example, all Ethernet interfaces start with **eth**, regardless of the specific hardware vendor. Following the prefix, each interface is numbered, starting at zero. For example, **eth0**, **eth1**, and **eth2** would refer to the first, second, and third Ethernet interfaces. Other interface names include **wlan0** for the first wireless device, **virbr0** for the internal bridge set up for virtual hosts, **bond0** for the first bonded network device, and so on.

Network Interface Configuration

/sbin/ip is used to show or temporarily modify devices, routing, policy routing, and tunnels.

```
[root@demo ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:00:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.250/24 brd 192.168.0.255 scope global eth0
    inet6 fe80::5054:ff:fe00:fa/64 scope link
        valid_lft forever preferred_lft forever
[root@demo ~]# ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
```

```
link/ether 52:54:00:00:00:fa brd ff:ff:ff:ff:ff:ff
RX: bytes  packets  errors  dropped overrun mcast
91449      520      0      0      0      0
TX: bytes  packets  errors  dropped carrier collsns
14020      99      0      0      0      0
[root@demo ~]# ip route
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.250 metric 1
default via 192.168.0.254 dev eth0 proto static
```



Note

ip -6 route shows the IPv6 routing table.

Hostname Resolution

The **hostname** command displays or temporarily modifies the system's fully-qualified hostname.

```
[root@demo ~]# hostname
demo.example.com
```

The *stub resolver* is used to convert hostnames to IP addresses or the reverse. The contents of the file **/etc/hosts** is checked first.

```
[root@demo ~]# cat /etc/hosts
192.168.0.250 demo.example.com demo # Added by NetworkManager
127.0.0.1 localhost.localdomain localhost
::1 demo.example.com demo localhost6.localdomain6 localhost6
```

If an entry is not found in that file the stub resolver looks for the information from a DNS nameserver. The **/etc/resolv.conf** file controls how this query is done:

- **nameserver:** the IP address of a nameserver to query. Up to three nameserver directives may be given to provide backups if one is down.
- **search:** a list of domain names to try with a short hostname. Both this and **domain** should not be set in the same file; if they are, the last instance wins. See **resolv.conf(5)** for details.

```
[root@demo ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 192.168.0.254
```

The **getent hosts hostname** command can be used to test hostname resolution.

Modifying Network Configuration

NetworkManager may be installed on Red Hat Enterprise Linux 6. It consists of a core daemon, a GNOME Notification Area applet that provides network status information, and graphical configuration tools that can create, edit and remove connections and interfaces.

To change a NetworkManager-managed eth0 interface from using DHCP to using a static IP address:

1. Right-click the NetworkManager icon in the top Panel and select **Edit connections...**
2. On the **Wired** tab, select **System eth0** and click the **Edit...** button
3. Select the **IPv4 Settings** tab
4. On the **Method** drop-down menu, change **Automatic (DHCP)** to **Manual**
5. Under **Addresses** click **Add** and enter the IPv4 address, netmask (in VLSN or CIDR notation), gateway router, and DNS server to use
6. **IMPORTANT:** make sure that **Connect automatically** is checked so the interface starts at boot (rather than when the user logs in), and **Available to all users** is checked so that it is available system-wide
7. Click **Apply** to apply your changes.

It is also possible to configure the network by editing interface configuration files. Interface configuration files control the software interfaces for individual network devices. These files are usually named **/etc/sysconfig/network-scripts/ifcfg-*<name>***, where *<name>* refers to the name of the device that the configuration file controls. The following are standard variables found in the file used for static or dynamic configuration.

Static	DHCP	Any
BOOTPROTO=static	BOOTPROTO=dhcp	DEVICE=eth0
IPADDR=192.168.0.250		ONBOOT=yes
PREFIX=24		HWADDR=52:54:00:00:00:FA
GATEWAY=192.168.0.254		NM_CONTROLLED=yes
DNS1=192.168.0.254		

Table 2.2. Configuration Options for **ifcfg** file



Note

If you need to configure static routes, the configuration is stored per interface in **/etc/sysconfig/network-scripts/route-*<name>***. Details can be found in the Red Hat Enterprise Linux Deployment Guide, see below.

/etc/sysconfig/network is used to specify the fully-qualified hostname and may specify a static default route if DHCP is not in use:

```
[root@demo ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=demo.example.com
GATEWAY=192.168.0.254
```

As we saw above, **/etc/resolv.conf** specifies the IP addresses of DNS servers and the search domain.



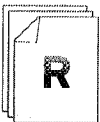
Important

If DHCP is in use, **/etc/resolv.conf** is automatically rewritten as interfaces are started unless you specify **PEERDNS=no** in the relevant interface configuration files.

Network interfaces can be brought down with the **ifdown eth0** command and brought back up with the **ifup eth0** command, whether managed by NetworkManager or by unmanaged configuration files.

When changing the system configuration you must remember to:

1. Modify a configuration file
2. Restart a service
3. Verify the change



References

Red Hat Enterprise Linux Deployment Guide
• Section 4.1: Network Configuration Files

Red Hat Enterprise Linux Deployment Guide
• Section 4.2: Interface Configuration Files

Red Hat Enterprise Linux Deployment Guide
• Section 4.4: Configuring Static Routes

Red Hat Enterprise Linux Deployment Guide
• Chapter 5: Network Configuration

/usr/share/doc/initscripts-*/sysconfig.txt

Network Troubleshooting Toolkit

Fill in how to TEST, CHECK, and FIX each of the network troubleshooting categories below from the list of commands following.

Category	TEST	CHECK	FIX
IP Address and Subnet Mask	ping Access a service	ip addr	Modify ifcfg-*
Routing/Default Gateway			
Name Resolution			

Table 2.3. Network Troubleshooting from the Command-Line

Useful commands and files

- **ping**

```
[root@demo ~]# ping -c 2 instructor.example.com
PING instructor.example.com (192.168.0.254) 56(84) bytes of data.
64 bytes from instructor.example.com (192.168.0.254): icmp_seq=1 ttl=64 time=0.697 ms
64 bytes from instructor.example.com (192.168.0.254): icmp_seq=2 ttl=64 time=0.538 ms

--- instructor.example.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.538/0.617/0.697/0.083 ms
```

- **ip addr show eth0**

```
[root@demo ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:00:00:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.250/24 brd 192.168.0.255 scope global eth0
    inet6 fe80::5054:ff:fe00:fa/64 scope link
    valid_lft forever preferred_lft forever
```

- **/etc/sysconfig/network-scripts/ifcfg-<name>**

```
[root@demo ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="dhcp"
```

```
HWADDR="52:54:00:00:00:FA"
NM_CONTROLLED="yes"
ONBOOT="yes"
```

• traceroute

```
[root@demo ~]# traceroute -Tn www.redhat.com
traceroute to www.redhat.com (184.85.80.112), 30 hops max, 60 byte packets
 1 192.168.0.254 0.641 ms 0.606 ms 0.590 ms
 2 172.31.35.1 9.829 ms 9.531 ms 9.237 ms
 3 204.60.4.40 27.954 ms 27.726 ms 27.385 ms
 4 66.159.184.226 27.128 ms 49.156 ms 48.291 ms
 5 151.164.92.147 43.256 ms 42.995 ms 42.155 ms
 6 12.122.81.57 60.897 ms 60.041 ms 54.531 ms
 7 75.149.230.169 54.143 ms 75.149.231.45 46.412 ms 192.205.37.34 40.208 ms
 8 68.86.86.45 67.587 ms 54.599 ms 53.381 ms
 9 68.86.86.234 65.540 ms 62.189 ms 53.777 ms
10 68.86.87.166 57.084 ms 55.752 ms 57.154 ms
11 184.85.80.112 55.707 ms 58.702 ms 57.996 ms
```

• ip route

```
[root@demo ~]# ip route
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.250 metric 1
default via 192.168.0.254 dev eth0 proto static
```

• host

```
[root@demo ~]# host i
i.example.com is an alias for instructor.example.com.
instructor.example.com has address 192.168.0.254
```

• dig

```
[root@demo ~]# dig i.example.com

;<<>> DiG 9.7.0-P2-RedHat-9.7.0-5.P2.el6 <<>> i.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 17644
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;i.example.com.                IN      A

;; ANSWER SECTION:
i.example.com.                 86400   IN      CNAME   instructor.example.com.
instructor.example.com.       86400   IN      A       192.168.0.254

;; AUTHORITY SECTION:
example.com.                   86400   IN      NS      instructor.example.com.

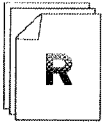
;; Query time: 2 msec
;; SERVER: 192.168.0.254#53(192.168.0.254)
;; WHEN: Mon Dec 13 15:50:21 2010
;; MSG SIZE rcvd: 86
```

- **/etc/hosts**

```
[root@demo ~]# cat /etc/hosts
192.168.0.250 demo.example.com demo # Added by NetworkManager
127.0.0.1 localhost.localdomain localhost
::1 demo.example.com demo localhost6.localdomain6 localhost6
```

- **/etc/resolv.conf**

```
[root@demo ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 192.168.0.254
```



References

Red Hat Enterprise Linux Deployment Guide

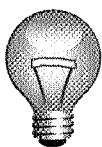
- Chapter 4: Network Interfaces

Red Hat Enterprise Linux Deployment Guide

- Chapter 5: Network Configuration

Network Interface Configuration - IP Aliases

Assigning multiple IP addresses to a single interface is called IP aliasing. This is useful for situations such as web hosting where a single machine may run different services or sites on different IP addresses. DHCP does not support aliases.



Important

This course recommends disabling NetworkManager when configuring aliases and bonding. NetworkManager supports multiple IP addresses in a way that is not backward compatible with old-style network alias configuration as done in Red Hat Enterprise Linux 5. NetworkManager also currently doesn't work with NIC bonding. It is expected that both of these limitations will be addressed in the future.

For more information about the new method of persistently configuring multiple IP addresses, consult <http://live.gnome.org/NetworkManager/SystemSettings#ifcfg-rh>.

There are three basic steps to adding an IP alias:

1. Persistently disable NetworkManager

```
[root@demo ~]# service NetworkManager stop ; chkconfig NetworkManager off
Stopping NetworkManager daemon: [ OK ]
```

2. Interactively add alias

```
[root@demo ~]# ip addr add 10.1.1.250/24 dev eth0 label eth0:0
[root@demo ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:00:fa brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.250/24 brd 192.168.0.255 scope global eth0
    inet 10.1.1.250/24 scope global eth0:0
    inet6 fe80::5054:ff:fe00:fa/64 scope link
        valid_lft forever preferred_lft forever
```

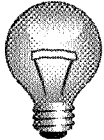
Persistently add alias by creating `/etc/sysconfig/network-scripts/ifcfg-eth0:0` with the following contents:

```
DEVICE=eth0:0
IPADDR=10.1.1.250
PREFIX=24
ONPARENT=yes
```

3. Restart the **network** service

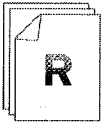
```
[root@demo ~]# service network restart
Shutting down interface eth0: [ OK ]
Shutting down loopback interface: [ OK ]
```

```
Bringing up loopback interface:      [ OK ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                     [ OK ]
```



Important

Avoid using the obsolete **ifconfig** command. If a system has new-style secondary IP addresses set on an interface that do not have a backward-compatibility IP alias label, **ifconfig** will not show the secondary address or addresses. Use the **ip addr** command instead.



References

Red Hat Enterprise Linux Deployment Guide

- Section 4.2.3: Alias and Clone Files

/usr/share/doc/initscripts-*/sysconfig.txt

Network Manager System Settings

<http://live.gnome.org/NetworkManager/SystemSettings#ifcfg-rh>

Network Interface Configuration - Bonding

Red Hat Enterprise Linux allows administrators to bind multiple network interfaces together into a single channel using the **bonding** kernel module and a special network interface called a channel bonding interface. Channel bonding enables two or more network interfaces to act as one, increasing bandwidth and/or providing redundancy, depending on the bonding mode chosen.

Physical Identification of a NIC

When working with multiple network cards, it is useful to be able to identify particular network cards physically. One method of physically identifying a NIC is to cause one or more of its LEDs to blink. To blink the LEDs on **eth0** for 30 seconds, run **ethtool -p eth0 30**.

Selected Linux Ethernet Bonding Modes

- Mode 0 (balance-rr) - Round robin policy, all interfaces are used. Packets are transmitted in a round-robin fashion through all slaves; any slave can receive.
- Mode 1 (active-backup) - Fault tolerant. Only one slave interface is in use at a time, but if it fails another slave takes over.
- Mode 3 (broadcast) - Fault tolerant. All packets are broadcast from all slave interfaces.

Other bonding modes are described in the kernel documentation **networking/bonding.txt** file.

Example Active-Backup Configuration

- **/etc/sysconfig/network-scripts/ifcfg-bond0**

This file configures the network information for the bonded interfaces, as if it were a normal network interface file:

```
DEVICE=bond0
IPADDR=10.1.1.250
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
BONDING_OPTS="mode=1 miimon=50"
```

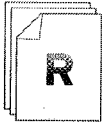
- **/etc/sysconfig/network-scripts/ifcfg-<name>**

Each slave interface *<name>* needs a file containing the following configuration:

```
DEVICE=<name>
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

- **/etc/modprobe.d/bonding.conf**

```
alias bond0 bonding
```



References

Red Hat Enterprise Linux Deployment Guide

- Section 4.2.2: Channel Bonding Interfaces

Red Hat Enterprise Linux Deployment Guide

- Section 22.7.2: Using Channel Bonding

`/usr/share/doc/kernel-*/Documentation/networking/bonding.txt`



Practice Quiz

Advanced Network Interface Configuration Quiz

1. Which mode of Linux Ethernet bonding primarily uses one slave interface and changes interface upon failure?

(select one of the following...)

- a. Mode 0 (balance-rr)
- b. Mode 1 (active-backup)
- c. Mode 3 (broadcast)

2. Which mode of Linux Ethernet bonding uses all interfaces in a round robin fashion to achieve more throughput?

(select one of the following...)

- a. Mode 0 (balance-rr)
- b. Mode 1 (active-backup)
- c. Mode 3 (broadcast)

3. When creating a bonded network interface, which configuration file contains the IP address and netmask definitions for the interface?

(select one of the following...)

- a. **/etc/sysconfig/network**
- b. **/etc/sysconfig/network-scripts/ifcfg-bond0**
- c. **/etc/sysconfig/network-scripts/ifcfg-iface**
- d. None of the above

4. When creating a bonded network interface, which configuration file defines the type of bonding?

(select one of the following...)

- a. **/etc/sysconfig/network**
- b. **/etc/sysconfig/network-scripts/ifcfg-bond0**
- c. **/etc/sysconfig/network-scripts/ifcfg-iface**
- d. None of the above

5. When creating a bonded network interface, which variable definitions must be specified in the **/etc/sysconfig/network-scripts/ifcfg-iface** configuration file?

(select one of the following...)

- a. **GATEWAY**
- b. **IPADDR**
- c. **MASTER**
- d. None of the above

Tuning Kernel Parameters

Kernel parameters provide a mechanism to adjust the functioning of the Linux kernel. Generally speaking, whenever a kernel developer selects an arbitrary constant or implements functionality that may not be generally desired, a **sysctl** will be made available to adjust it. Commonly useful parameters will be documented online, in **kernel-doc** or in this and other Red Hat courses.

These parameters can be viewed or set via the **/proc/sys/** directory tree or the **sysctl** command.

Kernel Tuning Search and Learn

The goal of this effort is to learn to tune how the kernel responds to ping, or ICMP echo, requests.

Investigate the **sysctl** command and scan the kernel documentation for relevant kernel parameters and answer the questions in your workbook. Write down the steps to complete the following tasks:

1. Install the **kernel-doc** RPM if it is not already installed.

```
[root@demo ~]# yum -y install kernel-doc
```

2. How would you use **sysctl** to identify kernel parameters that control ping, or ICMP echo, behavior?

```
[root@demo ~]# sysctl -a | grep icmp
```

3. Which parameters look promising?

net.ipv4.icmp_echo_ignore_all or **net.ipv4.icmp_echo_ignore_broadcasts**

4. What command would you use to identify and/or examine kernel documentation that describes what those parameters are for?

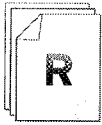
```
[root@demo ~]# grep -A5 icmp /usr/share/doc/kernel-doc-*/Documentation/networking/ip-sysctl.txt
```

5. How would you use **sysctl** to adjust kernel parameters to "hide" your system from ping requests?

```
[root@demo ~]# sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

6. How would you configure **sysctl** to persistently adjust kernel parameters to survive a reboot?

```
[root@demo ~]# echo "net.ipv4.icmp_echo_ignore_all = 1" >> /etc/sysctl.conf
```



References

Red Hat Enterprise Linux Deployment Guide

- Section 19.3.9.4: `/proc/sys/net/`

Red Hat Enterprise Linux Deployment Guide

- Section 19.4: Using the `sysctl` Command

`/usr/share/doc/kernel-doc-*/Documentation/sysctl/`

`/usr/share/doc/kernel-doc-*/Documentation/networking/ip-sysctl.txt`

`sysctl(8)` man page



Practice Performance Checklist

Enable Ping Broadcast

The default configuration for Red Hat Enterprise Linux 6 configures the kernel to ignore ping broadcast requests. You will work with a partner to tune the kernel on serverX to respond to them instead.

- ☐ Find a partner to work with. If there is an odd number of students, a group of three will work.
- ☐ Send a broadcast ping to the 192.168.0.0/24 network. Note which hosts respond to the ping request.

```
[root@serverX ~]# ping -b 192.168.0.255
```

- ☐ Tune serverX so that it will respond to ping broadcasts.

```
[root@serverX ~]# sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=0
```

- ☐ Send another broadcast ping to the 192.168.0.0/24 network. Did your hosts respond?

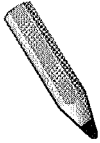
```
[root@serverX ~]# ping -b 192.168.0.255
```

- ☐ Persistently configure your serverX machines to respond to ping broadcasts and reboot.

```
[root@serverX ~]# echo "net.ipv4.icmp_echo_ignore_broadcasts = 0" >> /etc/  
sysctl.conf  
[root@serverX ~]# reboot
```

- ☐ Send another ping broadcast. Did your configuration changes persist the reboot?

```
[root@serverX ~]# ping -b 192.168.0.255
```



Test

Criterion Test 1

Case Study

Routing Network Traffic: Operation Strategic Holistic Unusual

Before you begin...

Run the **lab-setup-oshu** script on desktopX.

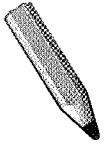
Operation Strategic Holistic Unusual (or OSHU), is an online chat system for fans of conspiracy fiction. In order to join the site they have two requirements, listed below.

1. To fulfill the first requirement, you must prove your ability to "disappear" a server. You will do this by modifying the configuration on serverX so that it does not respond to any ping requests. Make this change persistent so that it will still be in effect after a reboot.
2. The second requirement is to join the "secret" OSHU network. To join the network, add an additional IP address to serverX, where X is your desktop/server number:

10.42.10.X/24

When you have fulfilled the requirements, run **lab-grade-oshu** on desktopX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Test

Criterion Test 2

Exercise

Troubleshooting Network Configuration from the command-line

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of the following should be performed on your virtual server, serverX. You will start by running a script that will "break" your network configuration. You will have five minutes to resolve each of the two problems. Be sure to document what you have found, as we will review at the end.

The following are your network settings for serverX:

IP Address: 192.168.0.X+100
Netmask: 255.255.255.0 (/24)
DNS Server: 192.168.0.254
Default Gateway: 192.168.0.254

1. Run the first script to misconfigure your networking:

lab-break-net 1
2. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
3. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
4. Document what you have found

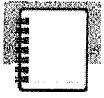
Use this space for notes



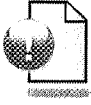
5. Run the second script to misconfigure your networking:

lab-break-net 2
6. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
7. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
8. Document what you have found

Use this space for notes



Personal Notes



Unit Summary

Understanding Network Configuration Files

In this section you learned how to:

- Change network configuration with command-line tools
- Make network configuration changes persistent by editing files

Network Troubleshooting Toolkit

In this section you learned how to:

- Troubleshoot basic network problems

Network Interface Configuration - IP Aliases

In this section you learned how to:

- Manually configure multiple IP addresses on one network card

Network Interface Configuration - Bonding

In this section you learned how to:

- Combine two network interfaces into one bonded interface

Tuning Kernel Parameters

In this section you learned how to:

- Make changes to kernel tuning parameters that affect networking settings



UNIT THREE

STORAGE MANAGEMENT

Introduction

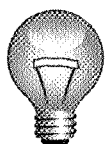
Topics covered in this unit:

- Creating and formatting simple disk partitions with a file system
- Enabling data privacy with an encrypted partition
- Creating and formatting a simple disk partition as swap space
- Connecting to and using a remote iSCSI target as storage

Simple Partitions and File Systems

Storage is a basic need of every computer system. Red Hat Enterprise Linux includes powerful tools for managing many types of storage devices in a wide range of scenarios.

fdisk is a utility to manage disk partitions. You can view disks and their partitioning by running the utility with the **-l** option and the name of the disk (**fdisk -cul /dev/vda**). Changes can be made by running the utility interactively and choosing appropriate menu options (**fdisk -cu /dev/vda**). **-c** disables legacy DOS-compatibility mode and **-u** displays output in sectors (not cylinders, which are obsolete).



Important

Red Hat Enterprise Linux 6 automatically aligns the first partition to start at sector 2048 instead of sector 63 (the "traditional" start of cylinder 1). This is to ensure maximum performance on new 4 KiB sector hard drives as well as legacy 512 byte sector hard drives, and is compatible with the behavior of other recent operating systems that use the MBR partitioning scheme. Partition misalignment can lead to significant performance loss, so be careful adjusting these settings.

For your virtual server, **serverX**, verify the current storage configuration. Look for information in the output of the following command: **fdisk -cul /dev/vda**

Primary Disk:

1. Name: **/dev/vda**
2. Size: **6442 MB**
3. Total sectors: **12582912**
4. Last used sector: **9914367**

```
[root@serverX ~]# fdisk -cul
```

```
Disk /dev/vda1: 6442 MB2, 6442450944 bytes
```

```
16 heads, 63 sectors/track, 12483 cylinders, total 12582912 sectors3
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x000a9b12
```

Device	Boot	Start	End	Blocks	Id	System
/dev/vda1	*	2048	526335	262144	83	Linux
/dev/vda2		526336	9914367 ⁴	4694016	8e	Linux LVM

- ¹ Name of disk
- ² Total size of disk
- ³ Total sectors

4 Last used sector

Create a new partition

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-12582911, default 9914368): Enter
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-12582911, default 12582911): +1G

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@serverX ~]# reboot
```

Filesystem Comparison

- **ext4** is the standard file system for Red Hat Enterprise Linux. It is very robust and reliable, and has many features to improve performance for modern workloads.
- **ext2** is an older file system commonly used in Linux; it is simple and reliable, and works well for small storage devices, but is not as efficient as **ext4**.
- **vfat** support covers a family of related file systems (VFAT/FAT16, FAT32) developed for older versions of Microsoft Windows and supported on a wide variety of systems and devices.

Creating and Using a New Filesystem

1. **mkfs -t filesystem /dev/partition** creates the type of file system requested.
2. **blkid** displays information about the contents of block devices (partitions and logical volumes) including the UUID of the filesystem.
3. **mkdir /mountpoint** creates a directory to link the new filesystem to.
4. Add an entry to **/etc/fstab** using the obtained UUID from the **blkid** command:

```
UUID=uuid /mountpoint ext4 defaults 1 2
```

5. Mount the new file system with **mount /mountpoint**.



Warning

When adding new file systems to **/etc/fstab**, you should use **blkid** to determine its' UUID and mount by UUID. You should *not* mount file systems on simple partitions by standard device name (such as **/dev/sda3**). Disk device names may change depending on the devices visible at boot time, which may cause your system to attempt to mount the wrong file system for the wrong purpose, which at worst could lead to data loss. This is especially important when SAN devices (iSCSI, Fiber Channel) are involved which may be detected by the system in a different order from boot to boot depending on SAN traffic, but it can also matter when removable media such as USB devices may be in use.

Note that Red Hat Enterprise Linux 6 uses UUID instead of LABEL in **/etc/fstab** to reduce the likelihood of naming collisions. The installer no longer uses **e2label** to set labels on RHEL 6 file systems by default.

Example of Filesystem Creation

```
[root@serverX ~]# mkfs -t ext4 /dev/vda3
[root@serverX ~]# blkid /dev/vda3
/dev/vda3: UUID="a11fadb0-2f5b-49e8-ba43-13de7990d3b9" TYPE="ext4"
[root@serverX ~]# mkdir /test
```

Add an entry to **/etc/fstab**:

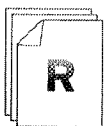
```
UUID="a11fadb0-2f5b-49e8-ba43-13de7990d3b9" /test ext4 defaults 1 2
```

Test the mount:

```
[root@serverX ~]# mount /test
```

Remove an Existing Filesystem

1. Unmount the filesystem by using **umount /mountpoint**.
2. Remove the corresponding entry in **/etc/fstab**.
3. Remove the mount point directory: **rmdir /mountpoint**.



References

fdisk(8), **fstab(5)**, **mkfs(8)**, **blkid(8)**, **partprobe(8)**, **mount(8)** man pages

Knowledgebase: "How can I create a disk partition on a disk that is greater than 2 TB in size?"

<https://access.redhat.com/kb/docs/DOC-4282>



Practice Quiz

Add a New Filesystem

1. Identify a disk that has some free space

2. Create a new partition on that disk

3. Update the kernel partition table

4. Create a filesystem on the partition

5. Add an entry to the filesystem table file

6. Create a mount point

7. Mount the filesystem

Enabling Data Privacy with Partition Encryption

LUKS ("Linux Unified Key Setup") is a standard format for device encryption. LUKS encrypts the partition or volume; the volume must be decrypted before the file system in it can be mounted.

Create a New Encrypted Volume

1. Create a new partition with **fdisk**
2. **cryptsetup luksFormat /dev/vdaN** will encrypt the new partition and sets the decryption password
3. **cryptsetup luksOpen /dev/vdaN name** unlocks the encrypted volume **/dev/vdaN** as **/dev/mapper/name** after you enter the correct decryption password.
4. Create an **ext4** filesystem on the decrypted volume: **mkfs -t ext4 /dev/mapper/name**
5. Create the directory mountpoint and mount the filesystem: **mkdir /secret ; mount /dev/mapper/name /secret**
6. When finished, **umount /dev/mapper/name** and run **cryptsetup luksClose name** to lock the encrypted volume

Persistently Mount Encrypted Partition

1. **/etc/crypttab** contains a list of devices to be unlocked during system startup.

1name **2**/dev/vdaN **3**/path/to/password/file

/etc/crypttab lists one device per line, with the following space separated fields:

- 1** Name device mapper will use for the device
- 2** The underlying "locked" device
- 3** Password file to use to unlock the device. If this field is left blank (or set to **none**), the user will be prompted for the decryption password during startup

2. Create an entry in **/etc/fstab** like the following:

```
/dev/mapper/name /secret ext4 defaults 1 2
```



Warning

The device listed in the first field of **/etc/fstab** must match the name chosen for the local name to map in **/etc/crypttab**. *This is a common configuration error.*

3. Create the key file that includes the password. Make sure it is owned by root and the mode is 600. Add the key for LUKS using the following command:


```
[root@serverX ~]# cryptsetup luksAddKey /dev/vdaN /path/to/password/file
```

Encrypted File System Creation Example

Create a new partition as previously done. We will assume the device is **/dev/vda5**.

```
[root@serverX ~]# cryptsetup luksFormat /dev/vda5
WARNING!
=====
This will overwrite data on /dev/vda5 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: testing123
Verify passphrase: testing123
[root@serverX ~]# cryptsetup luksOpen /dev/vda5 encdisk
Enter passphrase for /dev/vda5: testing123
[root@serverX ~]# mkfs -t ext4 /dev/mapper/encdisk
[root@serverX ~]# mkdir /encdisk
[root@serverX ~]# mount /dev/mapper/encdisk /encdisk
```

To make the disk persistent, start by appending the following to **/etc/fstab**:

```
/dev/mapper/encdisk /encdisk ext4 defaults 1 2
```

Create **/etc/crypttab** and add the following line. This will ask the password every time the machine boots:

```
encdisk /dev/vda5
```

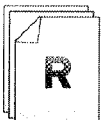
Automatic entry of encryption password

If you want an automated boot, you must place the password in a text file (which has obvious security implications).

/etc/crypttab:

```
encdisk /dev/vda5 /root/encdisk
```

```
[root@serverX ~]# echo "testing123" > /root/encdisk
[root@serverX ~]# chown root /root/encdisk
[root@serverX ~]# chmod 600 /root/encdisk
[root@serverX ~]# cryptsetup luksAddKey /dev/vda5 /root/encdisk
Enter any passphrase: testing123
```



References

cryptsetup(8) and **crypttab**(5) man pages



Practice Resequencing Exercise

Create Encrypted Filesystem

For each of the file or directory names below, write down the number of its definition from the list at the bottom.

- ___ Create a new partition
 - ___ Create an **ext4** filesystem
 - ___ Format the new partition for encryption
 - ___ Mount the filesystem on the unlocked device
 - ___ Create an entry in **/etc/fstab**
 - ___ Create a directory to use as a mount point
 - ___ Unlock the encrypted partition
 - ___ Create an entry in **/etc/crypttab**
 - ___ Make LUKS aware of the password file
-
- ___ 1. **fdisk**
 - ___ 2. **cryptsetup luksFormat /dev/vdaN**
 - ___ 3. **cryptsetup luksOpen /dev/vdaN secret**
 - ___ 4. **mkfs -t ext4 /dev/mapper/secret**
 - ___ 5. **mkdir /secret**
 - ___ 6. **mount /dev/mapper/secret /secret**
 - ___ 7. **secret /dev/vdaN /password/file**
 - ___ 8. **/dev/mapper/secret /secret ext4 defaults 1 2**
 - ___ 9. **cryptsetup luksAddKey /dev/vdaN /password/file**

Managing Swap Space

Swap space or a swap area is space on the disk drive used as overflow for parts of memory that are not currently being used. This allows the system to make room in main memory for data currently being processed, and provides emergency overflow if the system is at risk of running out of space in main memory.

Creating and Using an Additional Swap Partition

1. Create a new partition using **fdisk**. *Additionally*, change the partition type to "**0x82 Linux Swap**" before saving changes with **fdisk**.
2. **mkswap /dev/vdaN** will prepare the partition for use as a swap area.
3. **blkid /dev/vdaN** will determine the UUID.
4. Add the new swap space to **/etc/fstab**:

```
UUID=uuid swap swap defaults 0 0
```

5. **swapon -a** will activate the new swap area.

swapon -s will show status of current swap areas.

swapoff /dev/vdaN will de-activate that particular swap area.

Example of Swap Space Creation

Create a new partition and change the type to 82:

```
[root@serverX ~]# fdisk /dev/vda
Command (m for help): n
First sector (12539904-12582911, default 12539904): Enter
Using default value 12539904
Last sector, +sectors or +size{K,M,G} (12539904-12582911, default 12582911): Enter
Using default value 12582911
```

```
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap / Solaris)
```

```
Command (m for help): w
```

```
[root@serverX ~]# reboot
```

Write the swap signature to the device and find the UUID:

```
[root@serverX ~]# mkswap /dev/vda6
[root@serverX ~]# blkid /dev/vda6
/dev/vda6: UUID="4903c440-ffcb-4404-bc09-505c79c7a412" TYPE="swap"
```

Add an entry to **/etc/fstab**:

```
UUID="4903c440-ffcb-4404-bc09-505c79c7a412" swap swap defaults 0 0
```

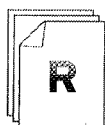
Activate the swap space, verify it is available and then deactivate the swap space:

```
[root@serverX ~]# swapon -a
swapon -s
/dev/dm-0                partition      557048 0      -1
/dev/vda6                partition      21496  0      -2
[root@serverX ~]# swapoff /dev/vda6
```

Sizing total swap space should really be based on the memory workload on the system, not the total amount of physical memory present. However, the table below provides some rough rules of thumb for sizing swap space. For more detailed guidance on sizing swap space, see the Knowledgebase article in the references.

System RAM	Recommended Minimum Swap Space
up to 4 GB	at least 2 GB
4 GB to 16 GB	at least 4 GB
16 GB to 64 GB	at least 8 GB
64 GB to 256 GB	at least 16 GB

Table 3.1. Basic Guidance on Swap Space Sizing



References

Knowledgebase: "If I add several hundred GB of RAM to a system, do I really need several hundred GB of swap space?"
<https://access.redhat.com/kb/docs/DOC-15252>

mkswap(8) and **swapon(8)** man pages



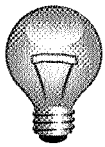
Practice Exercise

Create and use a new swap partition.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Create and use a new 256 MB swap partition on your virtual server, serverX.

1. Start **fdisk** and create a new partition

**Important**

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand

2. Change the partition type to **swap**.
3. Prepare the new partition for use as swap
4. Determine the UUID
5. Add the new partition to **/etc/fstab**
6. Determine current amount of swap
7. Activate the new swap
8. Verify newly activated swap

Accessing iSCSI Storage

iSCSI (Internet SCSI) supports sending SCSI commands from clients (initiators) over IP to SCSI storage devices (targets) on remote servers. An iSCSI Qualified Name is used to identify initiators and targets and follows the format of: **iqn.yyyy-mm.{reverse domain}:label**. Network communication by default is cleartext to port 3260/tcp on the iSCSI target.

- iSCSI initiator: a client that needs access to raw SAN storage
- iSCSI target: a remote hard disk presented from an iSCSI server, or "target portal"
- iSCSI target portal: a server that provides targets over the network to an initiator
- IQN: "iSCSI Qualified Name". Each initiator and target needs a unique name to identify it; best practice is to use one likely to be unique on the Internet.



Warning

If you allow two initiators to log in to the same iSCSI target (remote hard disk) at the same time, it is important not to allow both initiators to mount the same file system from the same target at the same time. Unless a cluster file system such as GFS2 is in use, you risk file system corruption.

To access a new target with an iSCSI initiator:

- Install iSCSI initiator software: *iscsi-initiator-utils*
- Set initiator's IQN in **/etc/iscsi/initiatorname.iscsi**

(Usually a unique label in a namespace matching a DNS name controlled by the organization. Set randomly when *iscsi-initiator-utils* is installed.)

- Discover iSCSI targets provided by the iSCSI server (target portal)

```
iscsiadm -m discovery -t st -p 192.168.0.254
```

- Log in to one or more iSCSI targets on the server

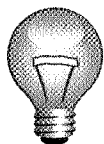
```
iscsiadm -m node -T iqn.2010-09.com.example:rdisks.demo -p 192.168.0.254 -l
```

- Identify which device is the iSCSI target

Look at the output of **dmesg** or **tail /var/log/messages**, or look at where the *iscsi* symlinks point with **ls -l /dev/disk/by-path/*iscsi***.

- At this point the iSCSI disk can be used as if it were a locally-attached hard drive.

Existing file systems can be mounted. If the disk is unformatted it can be partitioned with **fdisk**, and the partitions formatted with a filesystem or as an LVM physical volume, for example.



Important

When persistently mounting a file system on an iSCSI target in **/etc/fstab**:

1. Use **blkid** to determine the file system UUID and mount using UUID, not **/dev/sd*** device name. (The device name can come up differently from boot to boot depending on the order in which iSCSI devices respond over the network. This can cause the wrong device to be used if mounting by device name.)
2. Use **_netdev** as a mount option in **/etc/fstab**. (This ensures that the client will not attempt to mount the file system until networking is up. Otherwise the system will have errors at boot.)
3. Ensure the **iscsi** and **iscsid** services will start at boot time.

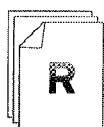
To discontinue use of an iSCSI target:

- Make sure none of the devices provided by the target are in use.
- Make sure all persistent references to use the target are removed from places like **/etc/fstab**.
- Log out of the iSCSI target to temporarily disconnect.

```
iscsiadm -m node -T iqn.2010-09.com.example:rdisks.demo -p 192.168.0.254 -u
```

- Delete the local record of the iSCSI target to persistently disconnect.

```
iscsiadm -m node -T iqn-2010-09.com.example:rdisks.demo -p 192.168.0.254 -o delete
```



References

Red Hat Enterprise Linux Storage Administration Guide

- Chapter 21: Online Storage Management

/usr/share/doc/iscsi-initiator-utils-*/README

Knowledgebase: "Can I put a swap device or file on iSCSI storage?"

<https://access.redhat.com/kb/docs/DOC-4135>

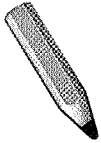


Practice Performance Checklist

Configuring iSCSI Exercise

You are configuring your serverX to use iSCSI storage existing on instructor.example.com.

- ☐ Log into serverX and become root.
- ☐ Verify that the *iscsi-initiator-utils* package is installed, and install it if needed.
- ☐ Discover iSCSI targets on the iSCSI server on 192.168.0.254.
- ☐ Log in to the iSCSI target **`iqn.2010-09.com.example:rdisks.serverX`** on 192.168.0.254.
- ☐ Identify the device file for your new iSCSI disk on your initiator.
- ☐ Set up a single partition on your new storage device, format the partition as ext4, and configure it to persistently mount on /mnt at boot. (*Note: Do not forget to use **`_netdev`** as a mount option, or to mount by file system UUID and not by standard device name.*)
- ☐ Test your configuration.
- ☐ Persistently unmount the new filesystem.
- ☐ Logout and delete the entry for the iSCSI target.



Test

Criterion Test

Exercise

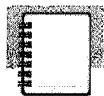
Partitions and Filesystems Lab

Before you begin...

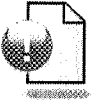
Reset serverX by running **lab-setup-server** from desktopX.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. On serverX, connect to the iSCSI target **iqn.2010-09.com.example:rdisks.serverX** from 192.168.0.254 and ensure it is enabled at boot time.
2. Create two new physical partitions using the iSCSI disk, 10 MB in size each.
3. With the first partition, create an ext4 filesystem mounted on **/test** and make it persistent.
4. With the second partition, create an ext4 file system persistently mounted on **/opt** with **acl** as a default mount option.



Personal Notes



Unit Summary

Simple Partitions and File Systems

In this section you learned how to:

- Create and format a simple partition for data storage

Enabling Data Privacy with Partition Encryption

In this section you learned how to:

- Enable data privacy with an encrypted partition from the command-line

Managing Swap Space

In this section you learned how to:

- Create and format a simple partition for swap

Accessing iSCSI Storage

In this section you learned how to:

- Access, format, and mount an iSCSI storage device
- Permanently disconnect from an iSCSI storage device



UNIT FOUR

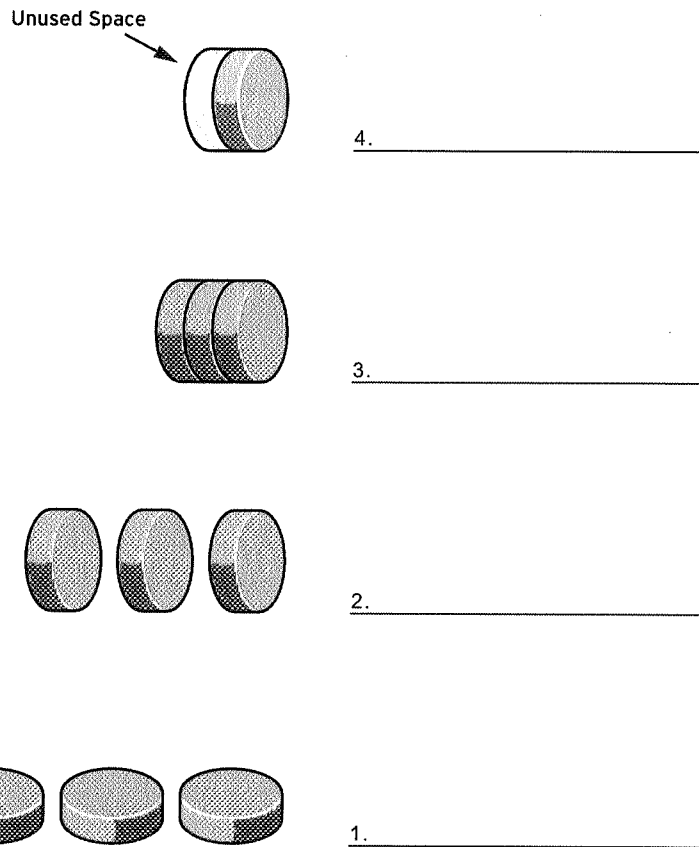
LOGICAL VOLUME MANAGEMENT

Introduction

Topics covered in this unit:

- Components of logical volume management
- Using LVM command-line tools
- Extending logical volumes and their ext4 file systems
- Adding a disk to a volume group
- Creating and using LVM snapshots

Recognize the Components of LVM



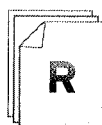
Review LVM Definitions

- *Physical Partitions or Disks* are the first building block of LVM. These could be partitions, whole disks, RAID sets or SAN disks.
- *Physical Volumes* are the underlying "physical" storage used with LVM. This is typically a block device such as a partition or whole disk. A device must be initialized as an LVM Physical Volume in order to be used with LVM.
- *Volume Groups* are storage pools made up of one or more Physical Volumes.
- *Physical Extents* are small chunks of data stored on Physical Volumes that act as the back end of LVM storage.
- *Logical Extents* map to Physical Extents to make up the front end of LVM storage. By default, each Logical Extent will map to one Physical Extent. Enabling some options will change this mapping. Mirroring, for example, causes each Logical Extent to map to two Physical Extents.

- *Logical Volumes* are groups of Logical Extents. A Logical Volume may be used in the same manner as a hard drive partition.

Why Use Logical Volumes?

Logical volumes, and logical volume management help make it easier to manage disk space. If a file system needs to have more space, it can be allocated to its logical volume from the free space in its volume group and the file system can be resized. If a disk starts to fail, a replacement disk can be registered as a physical volume with the volume group and the logical volume's extents can be relocated to the new disk.



References

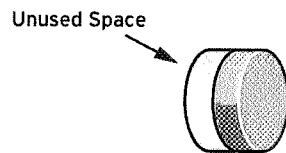
Red Hat Enterprise Linux Logical Volume Manager Administration Guide



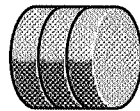
Practice Quiz

LVM Components

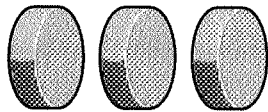
1. Fill in the following graphic with the names of the components.



4. _____



3. _____



2. _____



1. _____

2. What are the smallest pieces (chunks or blocks) of the physical volume?
3. What is the smallest size you could make a logical volume?
4. What references the physical extents of a logical volume?

Implement LVM Storage with Command-line Tools

Prepare a Physical Volume

1. **fdisk** is used to create a new partition for use with LVM. Always set the Type to **0x8e Linux LVM** on a partition to be used with LVM.



Note

Alternatively, you can use a whole disk, a RAID array, or a SAN disk.

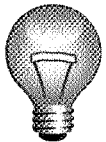
2. **pvccreate /dev/vdaN** is used to initialize the partition (or other physical device) for use with LVM as a Physical Volume. A header to store LVM configuration data is created directly in the Physical Volume.

Creating a Volume Group

1. **vgcreate vname /dev/vdaN** will create a volume group named *vname* made up of the physical volume */dev/vdaN*. You can specify additional space-delimited physical volumes at the time of creation or add new physical volumes later with **vgextend**.

Create and Use a New Logical Volume

1. **lvcreate -n lvname -L 2G vname** creates a new 2 GB logical volume named *lvname* from the available physical extents on *vname*.



Important

Different tools will display the logical volume name using either the traditional name, **/dev/vname/lvname**, or the kernel device mapper name, **/dev/mapper/vname-lvname**.

2. **mkfs -t ext4 /dev/vname/lvname** will create an **ext4** filesystem on the new logical volume.
3. **mkdir /data** makes directory needed as a mount point.
4. Add an entry to the **/etc/fstab** file:

```
/dev/mapper/vname-lvname /data ext4 defaults 1 2
```

5. **mount -a** mounts the filesystem now listed in **/etc/fstab**.

Review LVM Status Information

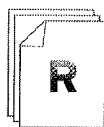
1. **pvdisk** **/dev/vdaN** will display information about the specific physical volume.
2. **vgdisplay** **vgname** will display information about the specific volume group.

Example **vgdisplay** output:

```
--- Volume group ---
VG Name ❶          vg1
System ID
Format    lvm2
Metadata Areas    10
Metadata Sequence No 18
VG Access    read/write
VG Status    resizable
MAX LV      0
Cur LV     6
Open LV     6
Max PV      0
Cur PV     5
Act PV     5

VG Size ❷          7.28 TB
PE Size ❸          4.00 MB
Total PE ❹         1907727
Alloc PE / Size ❺  1720587 / 6.56 TB
Free PE / Size ❻  187140 / 731.02 GB
VG UUID      7FmSCA-HJWa-<snip>
```

- ❶ The name of the Volume Group
 - ❷ The total size of the “physical” storage in the Volume Group
 - ❸ Physical Extent size
 - ❹ Total Physical Extents in Volume Group
 - ❺ Total Physical Extents used by Logical Volumes
 - ❻ Physical Extents available
3. **lvdisplay** **/dev/vgname/lvname** will display information about the specific logical volume.



References

Red Hat Enterprise Linux Logical Volume Manager Administration Guide

lvm(8) man page



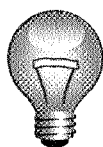
Practice Exercise

Implement LVM and Create a Logical Volume

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Create a new partition of 512 MB and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

2. Create a Volume Group named **shazam** using the Physical Volume created in the previous step.
3. Create and format with **ext4**, a new Logical Volume of 256 MB called **/dev/shazam/storage**.
4. Modify your system such that **/dev/shazam/storage** is mounted at boot time as **/storage**.

Extend a Logical Volume and ext4 Filesystem

One benefit of logical volumes is the ability to increase their size without experiencing downtime. Free physical extents in a volume group can be added to a logical volume to “extend” its capacity, which can then be used to extend the filesystem it contains.

Growing a Logical Volume Basic Steps

1. Verify available space in the

2. Extend the

3. Extend the

Extending the Logical Volume and Filesystem

1. Verify the current size of the mounted filesystem */data*:

```
# df -h /data
```

2. Verify there are sufficient "Physical Extents available" for use:

```
# vgdisplay vgname
```

3. Extend the logical volume using some or all of the available extents:

```
# lvextend -l +128 /dev/vgname/lvname
```

4. Grow the associated filesystem mounted on */data*:

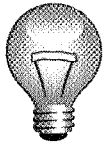
```
# resize2fs -p /dev/vgname/lvname
```

The **-p** option displays progress during the operation.



Note

The filesystem can remain mounted and be in use while **resize2fs** is being run.



Important

A common mistake is to run **lvextend** but to forget to run **resize2fs**.

5. Verify the new size of the mounted filesystem `/data`:

```
# df -h /data
```

Reducing a Filesystem and Logical Volume

This process is similar to extending, but *in reverse*: **resize2fs**, then **lvreduce**.



Warning

It is essential you have a solid backup before undertaking a reduction in the logical volume, as typographical errors in the command line can cause data loss.

1. While extending a logical volume can be done while the filesystem is in use, reducing an **ext4** filesystem must be done offline.

umount /data to unmount the filesystem you want shrink.

2. **fsck -f /dev/mapper/vgname-lvname** to verify that all filesystem data structures are clean prior to resizing.
3. **resize2fs -p /dev/mapper/vgname-lvname 512M** will resize the filesystem to be 512 MB, presuming that the logical volume is larger than 512 MB.

Note: If you omit the *size* from the **resize2fs** command, it defaults to the size of the logical volume, perfect for extending the logical volume like done previously.

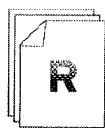
4. **lvreduce -L 512M /dev/mapper/vgname-lvname** will shrink the logical volume to 512 MB.



Warning

lvreduce has no knowledge of your filesystem data structures, and, without warning, will discard elements of your filesystem if you did not first use **resize2fs** to make the filesystem *smaller* than the intended logical volume size.

5. **mount -a** will remount your now smaller logical volume assuming it is listed in `/etc/fstab`.



References

Red Hat Enterprise Linux Logical Volume Manager Administration Guide

lvm(8) man page



Practice Exercise

Extend a Logical Volume

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Determine the amount of free space in Volume Group **shazam**.
2. Extend the logical volume **/dev/shazam/storage** with *half* the available extents in the volume group using command-line tools.
3. Extend the filesystem mounted on **/storage** using command-line tools.

Extending and Reducing a Volume Group

When the logical volumes in a volume group use all of the volume group's free physical extents, they cannot be extended without adding additional space to the volume group. Thankfully additional physical volumes can be created and added to a volume group to "extend" its capacity.

Another benefit of using LVM is that data can be moved between physical storage devices without user downtime. For example, data can be moved from a slower disk drive to a new, faster disk drive. This allows a system administrator to remove the unused physical storage device from a volume group, in this case the slow disk drive.

Extending a Volume Group

1. Similar to creating a new volume group, a new partition must be created and prepared for use as an LVM Physical Volume.

Use **fdisk** to create a new partition and set the Type to **0x8e Linux LVM**.

Use **pvccreate /dev/vdaN** to initialize the partition for use with LVM as a Physical Volume.

2. **vgextend vgname /dev/vdaN** is used to add the new Physical Volume, **/dev/vdaN**, to an existing Volume Group, **vgname**.
3. Use **vgdisplay** to confirm additional "Physical Extents available".

Reducing a Volume Group

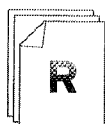
1. **pvmove /dev/vdaN** is used to relocate any physical extents used on **/dev/vdaN** to other Physical Volumes in the Volume Group. This is only possible if there are enough available extents in the Volume Group and if all of those come from other Physical Volumes.



Warning

Before using **pvmove**, it is recommended to back up data on logical volumes in the volume group. An unexpected power loss during the operation may leave the volume group in an inconsistent state.

2. **vgreduce vgname /dev/vdaN** is used to remove the Physical Volume **/dev/vdaN** from the Volume Group **vgname**.



References

Red Hat Enterprise Linux Logical Volume Manager Administration Guide

lvm(8), **pvmove(8)**, **vgreduce(8)** man pages



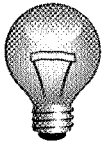
Practice Exercise

Extend a Volume Group

Carefully perform the following steps. Ask your instructor if you have problems or questions.

All of these steps will be performed on serverX.

1. Create a new 512 MB partition and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

2. Extend the Volume Group **shazam** by adding the Physical Volume created in the previous step.

Create a Snapshot to Facilitate Data Backup

Snapshot logical volumes are another flexible feature of LVM storage. An LVM snapshot is a logical volume that temporarily preserves the original data of a changing logical volume. The snapshot provides a static view of the original volume so its data can be backed up in a consistent state.

Determining Snapshot Size

1. Expected rate of _____
2. Required snapshot _____

The snapshot volume need only be large enough to store the data that will change while it exists.

If more data changes than the snapshot can hold, the snapshot will automatically become unusable. (The original volume will remain unharmed, and the dead snapshot will still need to be unmounted and removed from the volume group manually.)

Creating and Using a Snapshot for Backup

1. Create a new snapshot volume called **snaplvname** of **/dev/vgname/lvname** that is **20 MB** in size.


```
# lvcreate -s -n snaplv -L 20M /dev/vgname/lvname
```
2. If your backup software requires it, mount the snapshot and point the backup program to the new mountpoint:


```
# mkdir /snapmount  
# mount -ro /dev/vgname/snaplv /snapmount
```
3. Verify the status of the snapshot logical volume:


```
# lvs /dev/vgname/snaplv
```
4. When done with the snapshot, unmount and remove it:

```
# umount /snapmount  
# lvremove /dev/vgname/snaplv
```



References

Red Hat Enterprise Linux Logical Volume Manager Administration Guide

lvm(8) man page



Practice Exercise

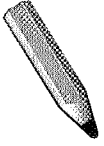
Creating an LVM Snapshot

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Compare the contents of our existing logical volume, **/dev/shazam/storage**, to a new snapshot volume, **/dev/shazam/storagesnap**, while making changes to the original volume.

All of these steps will be performed on serverX.

1. Copy the file **/usr/share/dict/linux.words** to **/storage** so you have some data to compare.
2. Create a new 20 MB snapshot logical volume of **/dev/shazam/storage** called **storagesnap**.
3. Manually mount **/dev/shazam/storagesnap** read only at **/storagesnap**
4. List the contents of **/storagesnap** and note that they are the same as **/storage**.
5. Delete the file **/storage/linux.words** and note that it still exists in **/storagesnap**.
6. Clean up: unmount **/storagesnap**, remove the directory, and delete the **storagesnap** logical volume.



Test

Criterion Test

Case Study

LVM Case Study

Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the lab.

Allison needs to store data for her business. Her customer database is currently 256 MB in size. The data in the database changes about 10 MB per hour on a typical day. The backup software takes 10 minutes to complete a full run.

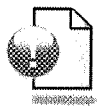
Create a new Volume Group called **allison** with enough space for both a 512 MB volume and a snapshot of that volume for the backup software. Create a 512 MB logical volume for Allison's customer database called **custdb**. Create a snapshot volume of Allison's customer database called **custdbsnap** for her backup software.

When you are ready, run the **lab-grade-lvm** script on serverX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Recognize the Components of LVM

In this section you learned how to:

- Identify the basic building blocks of Logical Volume Manager

Implement LVM Storage with Command-line Tools

In this section you learned how to:

- Create new physical volumes, volume groups and logical volumes via command-line tools
- Review LVM status information

Extend a Logical Volume and ext4 Filesystem

In this section you learned how to:

- Extend a logical volume and corresponding filesystem to satisfy growing data needs

Extending and Reducing a Volume Group

In this section you learned how to:

- Add new physical volumes to an existing volume group
- Remove an existing physical volume from a volume group

Create a Snapshot to Facilitate Data Backup

In this section you learned how to:

- Use temporary LVM snapshots to facilitate data backups, minimizing service downtime



UNIT FIVE

ACCOUNT MANAGEMENT

Introduction

Topics covered in this unit:

- Managing local user password aging policies
- Using extended ACLs to grant or block file access
- Setting extended ACLs on new files automatically

Managing Passwords

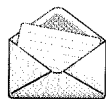
Historically, passwords were stored in **/etc/passwd**, but that file must be world readable to support username to UID mappings needed by utilities like **ls** to display the username rather than the UID number.

Passwords were migrated to a more secure **/etc/shadow** file where several different password encryption algorithms are supported. As long as encrypted passwords are being stored in a dedicated file, password aging policy and data can be stored as well.

What 3 pieces of information are stored in a password hash?

\$1\$gCjLa2/Z\$6Pu0EK0AzfCjxjv2hoLOB/

1. **1** - The hashing algorithm (1 indicates MD5 hash)
2. **gCjLa2/Z** - The salt used to encrypt the hash
3. **6Pu0EK0AzfCjxjv2hoLOB/** - The encrypted hash



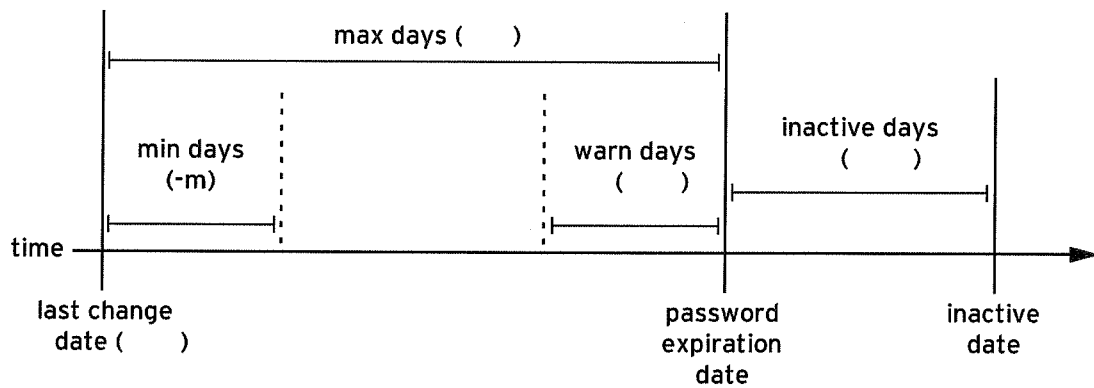
Note

Red Hat Enterprise Linux 6 supports two new strong password hashing algorithms, SHA-256 (algorithm **5**), and SHA-512 (algorithm **6**). These may be enabled as the default for **/etc/shadow** using **system-config-authentication** to select it from the **Password Hashing Algorithm** drop-down menu on the **Advanced Options** tab.

/etc/shadow Fields

1. Username
2. Password hash
3. Date of last password change (number of days since 1970.01.01)
4. Minimum password age (in days, 0 = no minimum age requirement)
5. Maximum password age (in days)
6. Password warning period (in days, 0 = no warning given)
7. Password inactive period (in days)
8. Account expiration (number of days since 1970.01.01)

The following diagram relates the relevant password aging parameters which can be adjusted using **chage** to implement a password aging policy.



As your instructor discusses these parameters, fill in the parenthesis in the above diagram with the relevant (short) **chage** command line switch.

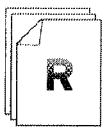
As an example, *-m* has been added to the *min days* parameter to get you started.

```
# chage -m 0 -M 90 -W 7 -I 14 username
```

chage -d 0 username will force a password update on next login.

chage -l username will list a username's current settings.

usermod can modify an account, including "locking" with the **-L** option.



References

Red Hat Enterprise Linux Deployment Guide

- Section 15.6: Shadow Passwords

chage(1), **shadow**(5), **crypt**(3) man pages



Practice Performance Checklist

Managing Password Aging Policies

Your instructor will divide you into small groups. Within each group, discuss which password aging policies would be appropriate for *professors* (who will be using the machine for a long time), *graduate students* (who will be using the machine for a few years), and *summer interns* (who will only be using the machine for the summer).

- **Professors:** faraday, juliet
- **Graduate Students:** jack, kate, james
- **Summer Interns:** walt, ben, clair, hugo
- ☐ If you do not already have the users and groups defined, run **lab-add-users** on serverX.
- ☐ For each group of users, determine a password aging policy which would be appropriate, including
 - Account expiration dates (if appropriate).
 - Time before passwords must be changed.
 - Time before unchanged passwords force an account to go inactive.
- ☐ Once determined, use **chage** to implement your policy for the users added in the previous section, according to their role.

Additionally, force all users to change their password on first login.

Managing Filesystem Access Control Lists

Access control lists provide for finer-grained security controls than the standard UGO (user, group, other) security scheme. They can be used to grant privileges, for example granting write privileges to a file. ACLs can also restrict privileges to a file, such as taking read access away from a specific member of a group that has group read access to a file.

Access Control List Support

- Standard Linux filesystems (ext2/3/4) support extended ACLs, provided they are mounted with the **acl** option.
- In Red Hat Enterprise Linux, if the last character of the permission string displayed by **ls -l** is a **+**, the file or directory has an ACL set.
- **getfacl file** is used to display ACLs on a file

```
u:elvis:rw-    # applies to user elvis
u:3142:---    # applies to user id 3142
u::rwx        # applies to file user owner

g:music:rwx    # applies to group music
g:10:r-x      # applies to group id 11
g::rw-        # applies to file group owner

o::rwx        # applies to everyone else
```

- **setfacl** is used to set or modify ACLs on a file

```
setfacl -m u:friend:rw filename # grants rw to user friend
setfacl -m g:grads:rw filename # grants rw to the group grads
setfacl -m g:profs:r filename  # grants r to the group profs

setfacl -x u:friend            # removes the existing ACL for friend

setfacl -m o::- filename       # changes normal "other" permissions
```

Use this space for notes

Permission Precedence

1. If process UID == file's user owner, use user permissions
2. If process UID == an explicit user ACL entry, use those (masked) permissions
3. If one of the groups of the process matches the group owner *or* an explicit group ACL entry, use *any* of those (masked) permissions which apply (in other words, it is cumulative across matching groups).

4. Otherwise, use the file's other permissions

Use this space for notes

Default ACLs and setgid Directories (Inheritance)

- Directories can have "default ACL" entries which are automatically set on new files created in that directory
- **setfacl -m d:u:elvis:rw directory** would set a default ACL entry granting read-write access to user **elvis** on all new files created in **directory**.
- This is similar to the way that the setgid permission, when set on a directory, causes new files created in that directory to be owned by the same group that owns the directory.

```
# chgrp staff /home/staff
# chmod 2775 /home/staff
# ls -ld /home/staff
drwxrwsr-x. 2 root staff 4096 2010-05-28 10:57 /home/staff
```

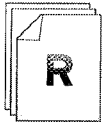
Use this space for notes

ACL Mount Option

- Support for extended ACL entries must be enabled when the file system is mounted.
- The installer configures all **ext4** file systems it creates to automatically turn on ACL support.
- If you manually formatted the file system, you will need to mount it with the **acl** mount option.
- You can set a manually-formatted **ext4** file system to turn on support at mount automatically by using **tune2fs** to set default mount options:

```
# tune2fs -o acl,user_xattr /dev/block-dev
```

Use this space for notes



References

Red Hat Enterprise Linux Storage Administration Guide

- Chapter 16: Access Control Lists

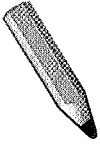
acl(5), **getfacl(1)**, **setfacl(1)** man pages



Practice Quiz

Collaborative Directory Permissions

1. What command would change the permissions on a directory to be a private single group collaborative directory?
2. What command would grant a second group access to that directory?
3. What command would grant that second group read-write access to any newly created files in that directory?



Test

Criterion Test

Exercise

Using ACLs to Grant and Limit Access

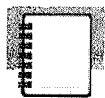
Carefully perform the following steps. Ask your instructor if you have problems or questions.

Using the users and groups created earlier on serverX....

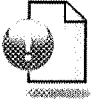
If you do not already have the users and groups defined, run **lab-add-users** on serverX.

Graduate Students need a directory /opt/research, where they can store generated research results. Newly created files in the directory should have the following properties:

1. The files should be group owned by the group grads.
2. Professors (members of the group profs) should have read/write access to the directory.
3. Summer interns (members of the group interns) should have read-only access to the directory.
4. Additionally, other users (not a member of profs, grads, or interns) should not be able to access the directory at all.



Personal Notes



Unit Summary

Managing Passwords

In this section you learned how to:

- Customize password aging policies for the users to meet organizational security requirements

Managing Filesystem Access Control Lists

In this section you learned how to:

- Use an ACL entry to grant or block file access
- List the ACLs on a file
- Delete an ACL entry
- Assign an ACL or group ownership to new files created in a directory automatically



UNIT SIX

AUTHENTICATION MANAGEMENT

Introduction

Topics covered in this unit:

- Configuring central authentication of LDAP-based users
- Configuring password authentication from a Kerberos server
- Diagnosing problems with sssd managed LDAP/Kerberos authentication
- Automatic mounting of NFS user home directories

Network Authentication Using an LDAP Server

So far in this class, we have looked at local user accounts managed through local files on each machine, `/etc/passwd`. But it is difficult to coordinate local user accounts to be the same on many systems.

In this section, we will look at how to set up a machine as a client, to use network user accounts that are provided by an existing LDAP directory service. This allows the LDAP directory to be our central authority for all network users and groups in our organization.

User account information determines the characteristics and configuration of the account. *Authentication methods* are used to determine if someone trying to log in should get access to the account. *Network directory services* can provide both user account information and authentication methods.

LDAP directory servers can be used as a distributed, centralized, network user management service. Directory entries are arranged in a tree structure that can be searched. The *base DN (Distinguished Name)* is the base of the tree that will be searched for directory entries for users and groups.

Key elements for LDAP client configuration

1. Server's fully-qualified hostname
2. Base DN to search for user definitions
3. The certificate authority ("CA") certificate used to sign the LDAP server's SSL certificate

Use this space for notes

You should ensure that the **directory-client** yum package group is installed, which includes the packages `sssd`, `authconfig-gtk`, and `oddjob-mkhomedir`, before you begin.

System → Administration → Authentication or **system-config-authentication** can be used to modify the configuration of *Identity & Authentication*.

The screenshot shows a window titled "Identity & Authentication" with a sub-tab "Advanced Options". It contains two main sections: "User Account Configuration" and "Authentication Configuration".

User Account Configuration

- User Account Database: LDAP (dropdown)
- LDAP Search Base DN: dc=example,dc=com (text box)
- LDAP Server: instructor.example.com (text box)
- ☒ Use TLS to encrypt connections
- Download CA Certificate... (button)

Authentication Configuration

- Authentication Method: LDAP password (dropdown)

Buttons at the bottom: Revert, Cancel, Apply.

The dialog box contains the following text: "To verify the LDAP server with TLS protocol enabled you need a CA certificate which signed the server's certificate. Please fill in the URL where the CA certificate in the PEM format can be downloaded from."

Certificate URL: <http://instructor.example.com/pub/EXAMPLE-CA-0>

Buttons: Cancel, OK.

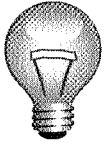
system-config-authentication will automatically turn on the **sssd** service which will look up and cache LDAP user information and authentication credentials for the client. If the LDAP server is unavailable but **sssd** is working, the system may be able to authenticate and get information about network users from the **sssd** cache.

Use **getent passwd username** to verify the account information being used. This works whether the user is a local user defined in **/etc/passwd** or a network user from an LDAP service. The command will always show the definition that is actually being used by the system if there is any duplication between local users and network users. By default, the local user definition overrides the network user definition.



Note

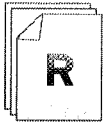
In Red Hat Enterprise Linux 6, **getent passwd** (without specifying a username) will only dump out local usernames by default, it will not dump out the list of all LDAP users as it did in Red Hat Enterprise Linux 5. This is done for performance reasons; see **sssd.conf(5)** under the **enumerate** option for details. (This behavior can be changed by setting **enumerate = True** in the **[domain/default]** section of **/etc/sss/sss.conf**.)



Important

When using **LDAP password** as your authentication method, you *must* select and configure **Use TLS to encrypt connections**. This is to prevent clear-text passwords from being sent to the LDAP server over the network for authentication.

This is a change from Red Hat Enterprise Linux 5, which would allow the insecure use of LDAP password authentication without TLS. In RHEL 6, you may still use LDAP without TLS if you are using LDAP to get user information only. (For example, you may be using Kerberos for password authentication.) It is better practice to always use TLS.



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 8: Authentication Configuration

system-config-authentication(8), **sssd(8)**, and **sssd.conf(5)** man pages



Practice Quiz

LDAP Client Configuration

1. What seven pieces of information are typically provided by *User account information* services?
2. What "other" type of information can be provided by a *network directory service*?
3. What are the three pieces of information a client machine needs to be configured to get user information from an LDAP directory service?
4. What does the command **getent passwd ldapuser1** do? Why is this useful?

Kerberos Configuration

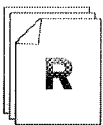
Kerberos is a method of secure authentication over an insecure network which was originally developed at MIT. Kerberos authenticates users without passing passwords over the network, protecting the integrity of the password and preventing its capture by network sniffers. Instead, it passes *tickets* which are encrypted using the user's password as an encryption key. When a user performs initial authentication to a system using a Kerberos password, the login program asks the Kerberos authentication server, or *key distribution center*, for a ticket for that user. The KDC sends the login program a ticket for that user, and if they type in the password that decrypts the ticket, the user has successfully authenticated.

Kerberos does not store

We will examine the steps and key configuration elements of pointing a system to an existing Kerberos realm.

Instead of using **system-config-authentication** to make changes, we will focus on the command-line tool, **authconfig**. Take notes on needed options with this tool as they are presented.

- Kerberos Realm - The set of machines that all use the same KDCs (Kerberos authentication servers) for authentication
- Key Distribution Center (KDC) - Central servers that store information about Kerberos passwords and issue Kerberos tickets (authentication credentials)
- Kerberos Admin Server - Servers that allow remote administration (for example, these servers are used to update passwords). Normally the master KDC is the admin server for the realm.



References

Red Hat Enterprise Linux Deployment Guide

- Section 8.1: The Authentication Configuration Tool

system-config-authentication(8), **kerberos(1)**, and **sssd(8)** man pages



Practice Performance Checklist

Kerberos Configuration Exercise

You will modify your previous LDAP-based configuration to now use only Kerberos for authentication. LDAP will still be used to provide account information.

- ☐ Log into serverX and escalate privileges to root
- ☐ Verify necessary packages are installed
- ☐ Configure system to use the following LDAP and Kerberos settings:
 - LDAP Server: instructor.example.com (uses TLS)
 - LDAP Certificate: ftp://instructor.example.com/pub/EXAMPLE-CA-CERT
 - LDAP Base DN: dc=example,dc=com
 - Kerberos Realm: EXAMPLE.COM
 - Kerberos KDC: instructor.example.com
 - Kerberos Admin Server: instructor.example.com
 - Be sure the **sssd** service is enabled
- ☐ Test the change by logging in to serverX with ssh:
 - Username: **ldapuserX** (where X is your station number)
 - Password: **kerberos**

Troubleshooting System Security Services Daemon (SSSD)

Authentication Troubleshooting

- As the instructor answers each of these questions, take notes in the space provided:
 1. What is the drawback of using LDAP or Kerberos to authenticate desktop or laptop users versus locally-defined user accounts?

 2. What can be implemented to resolve this problem?

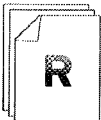
 3. How does one configure SSSD?

 4. What if I want to configure SSSD from the command line?

 5. How will this affect troubleshooting the authentication process?

 6. Where can we see what the SSSD service is doing?

 7. What if I cannot log in to view the log files or correct an authentication misconfiguration?



References

Red Hat Enterprise Linux Deployment Guide

- Section 8.2: The System Security Services Daemon (SSSD)

sssd.conf(5), **sssd-krb5**(5) **sssd-ldap**(5) man pages



Practice Quiz

Troubleshooting Authentication Quiz

1. How does one normally configure SSSD?

2. Which directory holds log messages from sssd?

3. How can we increase the logging detail that is generated?

4. When you cannot log in to correct an authentication misconfiguration, what approaches are available to you?

Network Mounting Home Directories

Recall that mounting network shares requires three pieces of information: sharename, mountpoint and mount options.

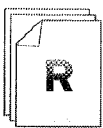
1. Use **showmount -e nfsserver.domain** to get the exported path that when combined with the hostname gives us the *sharename*.
2. Use **getent passwd username** to get the needed home directory *mountpoint*.
3. As home directories, we probably want to use **rw** as the *mount option*.

Configuring indirect maps in **autofs** would look something like this:

```
# cat /etc/auto.master
/home/guests /etc/auto.guests
# cat /etc/auto.guests
ldapuser1 -rw instructor.example.com:/home/guests/ldapuser1
ldapuser2 -rw instructor.example.com:/home/guests/ldapuser2
ldapuser3 -rw instructor.example.com:/home/guests/ldapuser3
ldapuser4 -rw instructor.example.com:/home/guests/ldapuser4
```

And, each time a new LDAP user is created, this file, **/etc/auto.guests** would need to be updated to include that additional user. However, notice the "pattern" to the lines. We want to support logging in as *any* username, so we could replace the first column with an "asterisk (*)", a wildcard, matching any subdirectory name that the login process may try to **cd** to. Then, we use the metacharacter, "ampersand (&)", to replace the username in the share which carries over the mapname matched by the wildcard:

```
# cat /etc/auto.master
/home/guests /etc/auto.guests
# cat /etc/auto.guests
* -rw instructor.example.com:/home/guests/&
```



References

Red Hat Enterprise Linux Storage Administration Guide

- Section 10.3: **autofs**

autofs(5), **auto.master(5)** man pages



Practice Performance Checklist

Use a NFS home directory server to provide automounted home directories.

The university also provides a NFS home directory server for its undergraduates. Use the NFS home directory server to automount the home directories of the previously defined users.

Here is information about the home directory server.

- Hostname: *instructor.example.com*

- Exported Directory: **/home/guests/**

- ☐ Extend the configuration of your automounter to mount to the **/home/guests** directory.

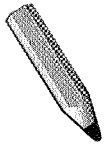
- ☐ Have the automounter attempt to map any specified target directory as the analogous directory from the home directory server.

As an example, a request to access the local directory **/home/guests/ldapuser1** should attempt to mount the directory **/home/guests/ldapuser1** from *instructor.example.com*.

- ☐ Have the automounter service reload its configuration files.

- ☐ From another terminal, attempt to shell into your remote server as the user **ldapuserX** with a password of **password**. The user's home directory should be automatically mounted.

- ☐ Run the **lab-grade-autofshomes** script when you are complete to verify your work.



Test

Criterion Test

Case Study

Enhance User Security

Before you begin...

Make sure to run the **lab-setup-taylorlocke** from your desktopX system, which will prepare your serverX system for the lab.

Taylor and Locke, a prestigious law firm, recently hired a security consultant to advise them regarding their servers. As the law firm's servers hold sensitive client information, security is a priority!

The security consultant recommended that all servers use LDAP for centralized accounts and Kerberos for authentication. Overall, the LDAP/Kerberos deployment went well. However, one of the servers that you manage appears to be mis-configured.

Correct the configuration on serverX so that LDAP users are able to login with Kerberos authentication (details below).

- LDAP Server: instructor.example.com (uses TLS)
- LDAP Certificate: ftp://instructor.example.com/pub/EXAMPLE-CA-CERT
- LDAP Base DN: dc=example,dc=com
- Kerberos Realm: EXAMPLE.COM
- Kerberos KDC: instructor.example.com
- Kerberos Admin Server: instructor.example.com

Test the change by logging in to serverX with ssh:

- Username: ldapuserX (where X is your station number)
- Password: kerberos

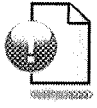
Once LDAP users can login, configure autofs to provide automounted home directories. The home directories are shared from instructor.example.com.

When you are ready, run the **lab-grade-taylorlocke** script on serverX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Network Authentication Using an LDAP Server

In this section you learned how to:

- Configure the system to authenticate users managed in a central LDAP directory service

Kerberos Configuration

In this section you learned how to:

- Use a Kerberos server to check user passwords

Troubleshooting System Security Services Daemon (SSSD)

In this section you learned how to:

- Correct problems with the **sssd** service and network authentication

Network Mounting Home Directories

In this section you learned how to:

- Automount existing NFS home directories for remote users using indirect map metacharacters



UNIT SEVEN

INSTALLATION, KICKSTART AND VIRTUALIZATION

Introduction

Topics covered in this unit:

- Creating a Kickstart file by modifying `/root/anaconda-ks.cfg` with a text editor
- Introduction to KVM virtualization
- Virtual guest installation
- Managing virtual machines

Creating a Kickstart File by Modifying a Template

Using *Kickstart*, a system administrator can create a single file containing the answers to all the questions typically asked during an installation. This file can then be accessed by the installer to automate installation of Red Hat Enterprise Linux.



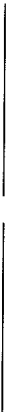
Comparison

Kickstart in Red Hat Enterprise Linux is similar to Jumpstart for Oracle Solaris, or to an unattended installation for Microsoft Windows.

Basic steps:

1. Create a Kickstart file
2. Make the Kickstart file available to the installer
3. Boot the installer
4. Point the installer to the Kickstart file

Use this space for notes



The Kickstart File

At the time of installation, the installer, **Anaconda**, will create **/root/anaconda-ks.cfg** containing the configuration settings used to install this system.



Note

Partitioning information is commented out in **/root/anaconda-ks.cfg** by default.

```
# Kickstart file automatically generated by anaconda.
```

```

#version=RHEL6
install
url --url=ftp://instructor.example.com/pub/rhel6/dvd
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$UaJVgaTh$KrpFf3K04r9hCZ2hsaa
# Reboot after installation
reboot
firewall --disabled
authconfig --useshadow --enablemd5
selinux --enforcing
timezone --utc America/New_York
bootloader --location=mbr --driveorder=vda --append="crashkernel=auto rhgb quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
#clearpart --all --drives=vda

#part /boot --fstype=ext4 --size=100
#part pv.ZS1CDM-iUYu-Gfua-YX0W-MSzd-ftBY-7qTB1E --size=28000
#part swap --size=512
#volgroup vol0 --pesize=32768 pv.ZS1CDM-iUYu-Gfua-YX0W-MSzd-ftBY-7qTB1E
#logvol /home --fstype=ext4 --name=home --vgname=vol0 --size=500
#logvol / --fstype=ext4 --name=root --vgname=vol0 --size=8192
repo --name="Red Hat Enterprise Linux" --baseurl=ftp://instructor.example.com/pub/rhel6/
dvd/ --cost=100

%packages
@Base
@Console internet tools
@Core
@Desktop
@Desktop Platform
@General Purpose Desktop
@Graphical Administration Tools
@Internet Browser
@Network file system client
@Printing client
@X Window System
lftp
mutt
ntp
%end

%post
# Turn on graphical login
perl -pi -e 's,id:3:initdefault,id:5:initdefault,' /etc/inittab
%end

```

Main sections of a Kickstart file

- Options that specify how the install should be done
- **%packages** (Package and yum group list)
- **%pre** (Script that runs before install starts)
- **%post** (Script that runs after install completes)

Reasons to manually edit a Kickstart file:

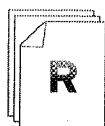
1. The GUI and/or **system-config-kickstart** is unavailable.
2. LVM instructions are needed.
3. Individual packages need to be included or omitted (not just groups).



Note

It is useful to check the syntax of your Kickstart file with **ksvalidator file.ks** before using it. This will check for gross typos and problems with the options; it does not validate the packages or groups list or the arbitrary **%pre/%post** scripts. **ksvalidator** is part of the *pykickstart* package.

Use this space for notes



References

Red Hat Enterprise Linux Installation Guide

- Chapter 32: Kickstart Installations



Practice Performance Checklist

Modify a Kickstart File without system-config-kickstart

Once you complete this exercise, you will have taken all of the steps necessary to Kickstart a new system, short of the actual installation. You will perform a Kickstart installation later in this unit. Perform the following steps on desktopX:

- ☐ Create a copy of `/root/anaconda-ks.cfg` called `~/projman.cfg`. Using only a text editor, modify that file so it meets the following criteria. The installation must be fully automated, and exactly like the basic workstation installation, except...
 - Perform the following disk partitioning:
 - Initialize the MBR if necessary
 - Clear all existing partitions
 - `/boot` (ext4) - 200 MB
 - swap - 512 MB
 - `/` (ext4) - all remaining space (5 GB minimum)
 - The **E-mail server** package group should be installed
 - The **fetchmail** package, which is not included with the **E-mail server** group by default, should be installed
 - Be sure to remove existing scripting from `%pre` and `%post`
 - Use **echo** to append the following text to the end of `/etc/issue`:

PROJECT MANAGEMENT
- ☐ **ksvalidator** must be able to validate the file
- ☐ When complete, publish the file so it can be used for an installation. Deploy a web server on desktopX and copy `projman.cfg` to `/var/www/html/`.
- ☐ Use a web browser to confirm your Kickstart file is readable. The URL you use to view the file is what you would pass to the installer with the **ks=URL** argument.

Introduction to KVM Virtualization

Virtualization is a feature that allows a single physical machine to be divided into multiple *virtual* machines, which can each run an independent operating system. Red Hat Enterprise Linux 6 for x86-64 supports *KVM*, which allows the kernel to function as a hypervisor supporting guest virtual machines, as long as certain requirements are met.

Facts about KVM Virtualization

- KVM = _____
 - Implemented as a kernel module
 - Allows a _____ Linux kernel to act as a hypervisor
- KVM requirements:
 - 64-bit AMD or Intel processors
 - _____ extensions
 - 64-bit operating system
- VirtIO support = _____ used by KVM
guests (provide better IO performance)
- KVM benefits include:
 - _____ performance
 - _____ design
 - _____ by upstream kernel developers

How to check if a machine supports KVM

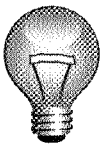
- _____
- Relevant flags include:
 - **lm** = _____ (64-bit x86)
 - **svm** = _____
(AMD)
 - **vmx** = _____
(Intel)



Note

Red Hat Enterprise Linux 6 can not act as a Xen hypervisor, although it can run as a para-virtualized or fully-virtualized Xen guest on a RHEL 5 Xen host. See *Red Hat Enterprise Linux Virtualization Guide* chapter 8, "Installing Red Hat Enterprise Linux 6 as a para-virtualized guest on Red Hat Enterprise Linux 5", for details.

Existing Xen guest machines from a Red Hat Enterprise Linux 5 host can be migrated to run as KVM guest machines on a Red Hat Enterprise Linux 6 host. See *Red Hat Enterprise Linux Virtualization Guide* chapter 23, "Migrating to KVM from other hypervisors using virt-v2v", for details.

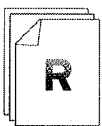


Important

There are two ways that the term *paravirtualization* is used in Linux virtualization which may lead to confusion.

In Red Hat Enterprise Linux 5, the Xen hypervisor supported *paravirtualized guests*. In this scenario, the drivers and kernel of the guests were modified to allow it to run on a Xen hypervisor running on a system that did not support full hardware virtualization extensions. This required that the operating system itself be modified to support Xen paravirtualized virtualization. KVM does not support paravirtualization in this sense.

KVM does support *paravirtualized drivers*. Paravirtualized drivers are special device drivers that can "cheat" by talking directly to the hypervisor. This removes the need for the guest to use a less efficient interface to the hypervisor that acts like some existing hardware device, like a disk controller or network card. These *virtio* paravirtualized drivers are faster than using normal drivers for the virtual hardware presented by KVM to the guest. Likewise, the operating system kernel does not need to be modified in order to take advantage of paravirtualized devices, you only need new drivers to be written which supports them.



References

Red Hat Enterprise Linux Virtualization Guide

- Chapter 5: Installing the virtualization packages



Practice Quiz

Introduction to KVM Virtualization

1. Hardware-assisted virtualization requires a special CPU with virtualization enabled in the BIOS.
(select one of the following...)
 - a. True
 - b. False

2. KVM is a kernel-based virtualization technology that allows both Linux and Windows to be installed as a virtual machine without using a special kernel.
(select one of the following...)
 - a. True
 - b. False

3. KVM is becoming popular because it achieves high performance because of its complex design.
(select one of the following...)
 - a. True
 - b. False

4. The **lm** and either **svm** or **vmx** CPU flags are required for kernel-based virtualization.
(select one of the following...)
 - a. True
 - b. False

5. KVM will work on both 32-bit and 64-bit hardware.
(select one of the following...)
 - a. True
 - b. False

6. Upstream software developers have adopted KVM into the source code for the Linux kernel.
(select one of the following...)
 - a. True
 - b. False

Virtual Guest Installation

When installing a virtual machine, there are several elements that must be chosen before proceeding with the rest of the installation via **Anaconda**.

Virtual Machine Specifications

1. A domain name must be specified
2. Point to install media to grab the 1st and 2nd stages of anaconda
3. Virtual hardware elements must be specified:
 - Number and types of CPU
 - Size of RAM
 - Virtual disk device (file or volume?)
 - Network connection and MAC address

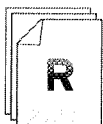
Virtual machines can be installed, managed, and accessed with **virt-manager**, a graphical tool. The instructor will demonstrate how to use **virt-manager** in class before you use it in the next practice exercise.



Note

Para-virtualized hard disks (that use the virtio drivers) appear to guests as **/dev/vd*** instead of **/dev/sd***.

Use this space for notes



References

Red Hat Enterprise Linux Virtualization Guide

- Chapter 6: Virtualized guest installation overview

Red Hat Enterprise Linux Virtualization Guide

- Chapter 7: Installing Red Hat Enterprise Linux 6 as a virtualized guest

virt-manager(1) man page



Practice Performance Checklist

Virtual Guest Installation

In this lab you will install a new virtual machine with Red Hat Enterprise Linux using **virt-manager** and the graphical installer. Once you have successfully completed the lab you will need to remove both the virtual machine and its logical volume to reclaim system resources needed for other labs.

Perform the following steps on desktopX:

- ☐ Gracefully shutdown your serverX virtual machine to reclaim system CPU and RAM resources.
- ☐ Create a logical volume 10 GB in size from the **vol10** volume group and name it **guest**.
- ☐ Create a Red Hat Enterprise Linux 6 virtual machine with the following characteristics:
 - Name = guest
 - Install media = network install from <http://instructor.example.com/pub/rhel6/dvd>
 - Memory (RAM) = 768 MB
 - CPUs = 1
 - Storage device = the logical volume created in the previous step
 - Network - use DHCP to get IP address
- ☐ Once **Anaconda** launches, build your guest system according to the following specifications:
 - Use the entire virtual drive with a default disk partitioning scheme
 - Assign **redhat** as the root password
 - Install the Desktop package group
- ☐ Reclaim the system resources used by this lab exercise. Remove the virtual machine you created and the storage it uses.

Manage Virtual Machines

Commands Used to Manage Virtual Machines

Until now **virt-manager** has been used to manage virtual machines. There is a command-line tool, **virsh**, that implements the same functionality as **virt-manager** without the need for a GUI. Both of these utilities use the **libvirt** library so they can be used interchangeably to manage virtual machines.

1. Power on a virtual machine: **virsh** _____
2. Gracefully shut down a virtual machine: **virsh** _____
3. Power off a virtual machine: **virsh** _____
4. Connect to a console of a virtual machine: **virsh** _____
5. Disconnect from a console of a virtual machine: _____
6. Start a virtual machine at boot time: **virsh** _____

Use this space for notes



References

Red Hat Enterprise Linux Virtualization Guide

- Chapter 30: Managing guests with **virsh**

virsh(1) man page

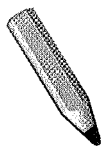


Practice Performance Checklist

Virtualization Commands

Perform all of the following tasks from the command line on desktopX. Do not use **virt-manager** or **virt-viewer** during this lab exercise.

- ☐ Use **virsh list --all** to determine the virtual domain ID (or name) of serverX. You will need the domain name to perform the following steps.
- ☐ If serverX is not running, power it on.
- ☐ Gracefully shut down serverX.
- ☐ Power on serverX.
- ☐ Connect to the console of serverX.
- ☐ The virtual machine may not be configured to present a console on the virtual console. Disconnect from the console.
- ☐ Power off serverX.
- ☐ Ensure serverX starts at boot time.



Test

Criterion Test

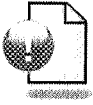
Performance Checklist

Kickstart a Virtual Machine

- ☐ Copy the **/root/anaconda-ks.cfg** file from serverX to desktopX and call it **~/test.cfg**. Shutdown serverX after you have copied the file to reclaim system resources for the rest of the lab.
- ☐ Modify **test.cfg** according to the following criteria:
 - Partition storage according to the following:
 - /boot (ext4) 200 MB
 - swap 512 MB
 - / (ext4) 8 GB
 - Add the **gimp** package
 - Create a **/root/install-date** file with the date and time.
- ☐ Copy **test.cfg** to **/var/www/html/** on desktopX. Make sure the file is readable by Apache. Start the **httpd** daemon if it is not already running.
- ☐ Create a logical volume in the volume group **vol0** named **test** large enough to serve as the disk for your virtual machine .
- ☐ Start a virtual machine installation using your **test.cfg** Kickstart file. Name the virtual machine **test**. Use the install media from <http://instructor/pub/rhel6/dvd> and allocate the virtual machine to have 768 MB of RAM and 1 CPU. Use the logical volume you created in the previous step as the storage for your virtual machine.
- ☐ Reboot your virtual machine when it is finished installing and confirm that it installed correctly.
- ☐ **IMPORTANT:** Delete your virtual machine and the logical volume it uses for storage to reclaim resources needed in future labs.



Personal Notes



Unit Summary

Creating a Kickstart File by Modifying a Template

In this section you learned how to:

- Modify an existing Kickstart configuration with the a text editor

Introduction to KVM Virtualization

In this section you learned how to:

- Describe the basic function, components, and benefits of KVM virtualization

Virtual Guest Installation

In this section you learned how to:

- Install a virtual guest according to specification

Manage Virtual Machines

In this section you learned how to:

- Manage virtual machines using the command-line **virsh** tool



UNIT EIGHT

BOOT MANAGEMENT

Introduction

Topics covered in this unit:

- Resolving issues with the GRUB bootloader
- Making persistent changes to the GRUB bootloader configuration
- Changing the default runlevel
- Using single-user mode to fix boot problems
- Troubleshooting problems in the boot process
- Using the installer's rescue environment
- Repairing common boot-time problems

Resolve GRUB Issues

The GRand Unified Bootloader (GRUB) provides the bridge in the boot process between the hardware and the Linux kernel. When the system boots, the BIOS starts and normally loads GRUB in stages from the hard drive; from the first 446 bytes of the disk, then from the space between the first sector and the start of the first partition, then from files in **/boot**. GRUB then reads its configuration file, **/boot/grub/grub.conf**, which controls what operating systems and kernels are available to boot.

The GRUB Boot Screen

When GRUB starts up, a graphical splash screen can be accessed by pressing Return, Space or any other key. This screen has a list of menu entries, normally bootable images. You can select between the different images with the up and down arrow keys, and press Return to select a particular entry for booting. If you want to pass arguments to boot images through menu editing mode or access the GRUB command line, and a GRUB password is set, you will need to type "p" followed by your GRUB password.

Temporary GRUB Correction

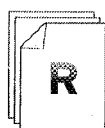
1. Interrupt the GRUB countdown: **Esc** key
2. Use "e" to edit current configuration
3. Select lines to correct with arrow keys
4. Type "e" again to edit the current line



Note

The **Esc** takes you back to the initial menu and throws changes away

5. The "b" command boots with the current changes



References

- Red Hat Enterprise Linux Installation Guide
- Technical Appendix E.5: GRUB Interfaces



Practice Performance Checklist

Resolve GRUB issues

- ☐ Run the **lab-setup-bootbreak** script on desktopX to prepare your virtual server for boot time problems.
- ☐ After serverX has booted, run the **lab-setup-bootbreak-5** script on serverX to introduce an issue with its boot process.
- ☐ Reboot serverX and modify the bootloader temporarily so the system can boot and you can log in.

Making Persistent GRUB Changes

The second stage of GRUB uses `/boot/grub/grub.conf` which has a format of global options followed by boot stanzas. Here is a sample `grub.conf` file:

```
[root@demo ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vgsrv-root
#          initrd /initrd-[generic-]version.img
#boot=/dev/vda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux (2.6.32-71.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-71.el6.x86_64 ro root=/dev/mapper/vgsrv-root
    rd_LVM_LV=vgsrv/root rd_LVM_LV=vgsrv/swap rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
    SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us crashkernel=auto rhgb quiet
    initrd /initramfs-2.6.32-71.el6.x86_64.img
```

- Comment lines begin with a `#` character
- **default=number** - *number* is the default boot stanza (starting from 0)
- **timeout=number** - specifies how long the countdown occurs
- **hiddenmenu** - hides the menu display until a key is struck
- **rhgb quiet** - consider removing these kernel arguments to view more diagnostic information during boot



References

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E.7: GRUB Menu Configuration File

Red Hat Enterprise Linux Deployment Guide

- Section 23.6: Verifying the Boot Loader

info grub



Practice Performance Checklist

Resolve GRUB issues persistently

- ☐ Reboot and ensure the issue from the previous problem is persistent. You will need to apply the fix as before to boot the system.
- ☐ Edit the configuration file to fix the issue permanently.
- ☐ Install a new kernel from the **Errata** repository.
- ☐ Revert to the older kernel. In other words, with the new kernel still available, ensure that when you reboot, the older kernel is the default kernel.
- ☐ Reboot the system to confirm that the old kernel successfully boots and you are able to log in.

Changing the Default Run Level

The runlevel determines which services are started automatically on your Linux system. Most Linux desktop systems are set to boot to runlevel 5 (multi-user, networking, graphical interface). Many server systems boot to runlevel 3 (multi-user, networking, no graphical login), where the system comes up to a text-based interface.

The command **who -r** will return the runlevel the system is currently using, as will the right-hand number in the output of **runlevel**.

The default runlevel is read from the **/etc/inittab** file. For example, the line below would cause the system to boot to runlevel 5 by default.

```
id:5:initdefault:
```

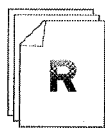


Note

In Red Hat Enterprise Linux 6, the new **Upstart** boot system is configured to read the default runlevel from **/etc/inittab** for backward compatibility purposes. None of the other services formerly controlled from that file, including login prompts, can be set up in that file in RHEL 6. Those settings are kept in **/etc/init/** directory instead, and how this works will be discussed in more detail later in this unit.

Changing Runlevels

- Execute **init rlnum** at the shell prompt, where *rlnum* is the runlevel number. This will change the runlevel immediately.
- Pass the runlevel number as an argument to the kernel via GRUB at boot time. This will override the default runlevel.



References

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E.8: Changing Runlevels at Boot Time

Comments in **/etc/inittab**



Practice Performance Checklist

Change the default runlevel

You are configuring a new system that you will be accessing remotely. The system is currently booting into run level 5 by default, but this machine will be housed in a data center where you will only login to it remotely. You want to change the serverX system to boot to runlevel 3 by default.

- ☐ Configure the system to boot into runlevel 3 by default.
- ☐ Reboot, then verify the current runlevel.

Single-User Mode

Single-user mode is a special runlevel which stops the boot process just before system services are started and opens a shell prompt as root. It is useful for troubleshooting purposes if a system is hanging during service startup but otherwise boots. This can happen due to misconfiguration of a system service or, in some cases, networking.

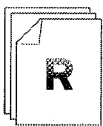
To boot into single-user mode, pass the argument **single** to the kernel instead of a runlevel number on the kernel command-line in the GRUB menu.



Note

In Red Hat Enterprise Linux 6, when booting into single-user mode the **init** process reads its **/etc/init/rcS.conf** file normally, which runs **/etc/rc.d/rc.sysinit**. However, then **init** reads **/etc/init/rcS-sulogin.conf** (which pauses the boot process and opens the root shell prompt), rather than reading **/etc/init/rc.conf** and running the service startup scripts. (See the next section for a diagram of the Upstart **init** boot process.)

In Red Hat Enterprise Linux 5 and earlier, the boot mechanism is slightly different, but single-user mode is started in the same way, and pauses the boot after running **/etc/rc.d/rc.sysinit** just before service startup scripts are normally run.



References

- Red Hat Enterprise Linux Installation Guide
- Section 36.1.3: Booting into Single-User Mode



Practice Performance Checklist

Changing the root Password

This timed drill is designed to give you practice changing the root password on a system with an unknown root password.

- ☐ Begin by running the **lab-setup-bootbreak-4** script on serverX. This will change the root password to something unknown and mark the current time.
- ☐ Get into the system and reset the root password to **redhat**.



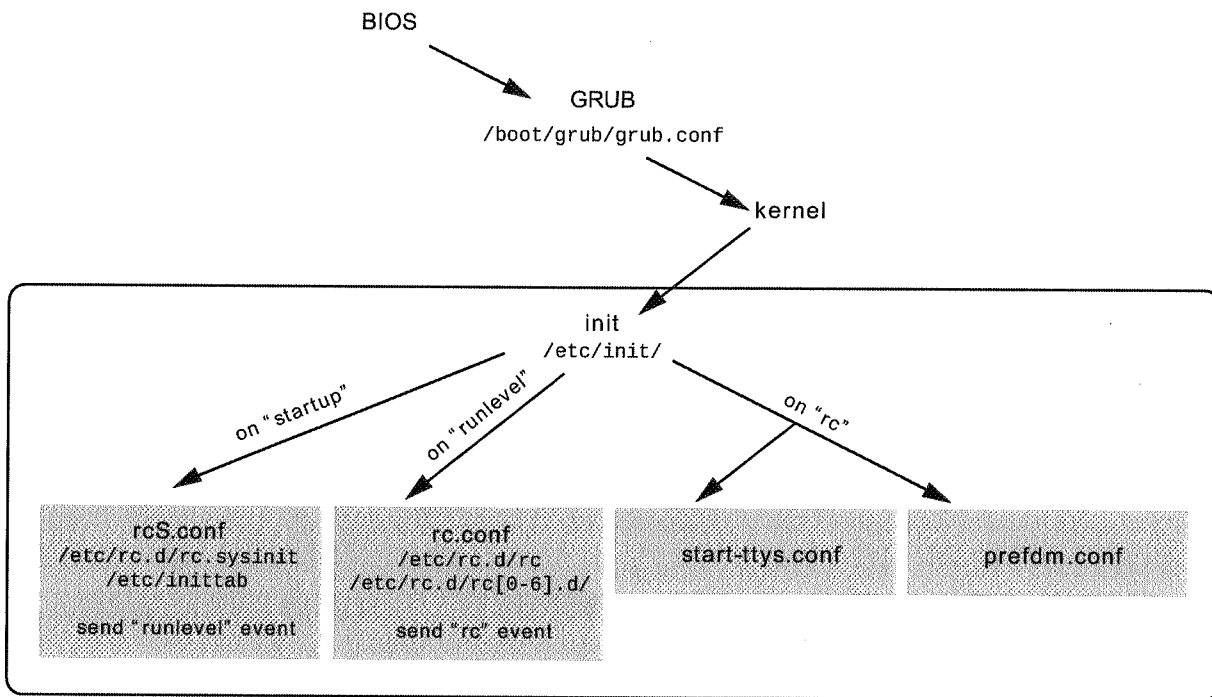
Note

At the release of Red Hat Enterprise Linux 6, there was an SELinux bug which blocked the **passwd** command in single-user mode (#644820). If you have the original *selinux-policy* package installed, you must run the **setenforce 0** command in runlevel 1 before the **passwd** command for it to work. After changing the password you should run **setenforce 1** again to put SELinux back in enforcing mode.

- ☐ Once you have reset the password, change the system into runlevel 5 and run the **lab-grade-bootbreak-4** script on serverX.
- ☐ View the feedback from the script to ensure you completed the task correctly. The grading script will display a time, write it down.
- ☐ Repeat the process again at least five times.
- ☐ Circle your best time.

The Boot Process and Rescue Mode

Below is a simplified diagram of the Red Hat Enterprise Linux 6 boot process, from power-on to the point at which a login prompt appears on the screen.



BIOS

The BIOS, or Basic Input/Output System, is the firmware interface built into standard x86/x86-64 hardware that puts the hardware into a known state and prepares the system to load an operating system.

What happens?

- Detects and initializes hardware
- Determines the device to boot from

What can go wrong?

- Incorrect or weird BIOS settings
- Bad boot device ordering

How can it be interrupted or influenced?

- Press a vendor-specific key
- Use a vendor-specific configuration utility
- Often, <F12> can perform a one-time override of the boot ordering

GRUB

GRUB, the GRand Unified Bootloader, is loaded by the BIOS and used to select and start the operating system, as we have already discussed.

What happens?

- Loads initial RAM filesystem ("initramfs")
- Loads and executes kernel
- Provides kernel's command-line

What can go wrong?

- Misconfiguration of the bootloader
- Bad kernel image or initramfs
- Bad kernel command line

How can it be interrupted or influenced?

- Choose an alternate pre-configured menu item
- Use "e" or "a" to select a different kernel image or edit the kernel command-line
- Edit the kernel command-line to boot from **single** user mode
- Boot with **init=/bin/bash**

Kernel

The Linux kernel is the heart of the operating system. It is responsible for managing hardware access for userspace processes. Drivers and the KVM hypervisor are integrated parts of the kernel.

What happens?

- Detect hardware devices
- Load device drivers (modules) for the devices



Note

Where does the kernel get modules to load at boot time?

1. Initially, it uses the initial RAM disk configured for the kernel in **/boot/grub/grub.conf: /boot/initramfs-<VERSION>.img**
 2. Once the root filesystem is mounted, it uses **/lib/modules/<VERSION>/**
- Mount the root filesystem read-only

- Start the initial process, **init**

What can go wrong?

- Bad initial RAM filesystem image
- Badly identified root filesystem
- Corrupted root filesystem

How can it be interrupted or influenced?

- Generally only through GRUB options

init and Upstart

The first userspace process started on the machine is **/sbin/init**. The **init** process is responsible for starting all remaining userspace processes, directly or indirectly.

What happens?

- Once the kernel is running, it starts **init**. The **init** program is responsible for completing the boot process by starting all other non-kernel system processes.

With **Upstart**, **init** starts "jobs" when various "events" happen, such as when the system boots, we enter a runlevel, or another **init** job starts or stops. These jobs are stored as scripts in the **/etc/init/** directory. At boot, the startup event causes **init** to run the **/etc/init/rcS.conf** job which:

- Runs **/etc/rc.d/rc.sysinit** to start LVM, mount and check filesystems, set the system clock, and do other housekeeping.
- Looks in **/etc/inittab** to find the runlevel.
- Sends an event to **init** telling it to enter that runlevel.

The runlevel event causes **init** to run the **/etc/init/rc.conf** job which runs the **/etc/rc.d/rc** script with the desired runlevel as an argument:

- Example: **rc.conf** runs **rc 5**, which runs **/etc/rc.d/rc5.d/K* stop** and **/etc/rc.d/rc5.d/S* start**.
- The scripts are run in numeric order, first the K's and then the S's.
- The **/etc/rc.d/rc5.d/** scripts are symlinks to the scripts used by service.
- Whether the links start with a K or an S depends on whether the service has been turned **on** or **off** with **chkconfig**.

Older versions of Red Hat Enterprise Linux have a somewhat different implementation of **init** which works differently.

What can go wrong?

- Mangled **/etc/fstab** or **/etc/crypttab**
- Network or service misconfiguration leading to hung services

- Many more...

How can it be interrupted or influenced?

- Press "Alt-D" from graphical environment to view error messages
- Press "I" (capital i) during service startup to select services interactively



Note

Red Hat Enterprise Linux 5 and earlier used a different implementation of **init**, **SysVinit**, that was driven by directives in the **/etc/inittab** file. The basic boot process and scripts run by **init** were similar, however.



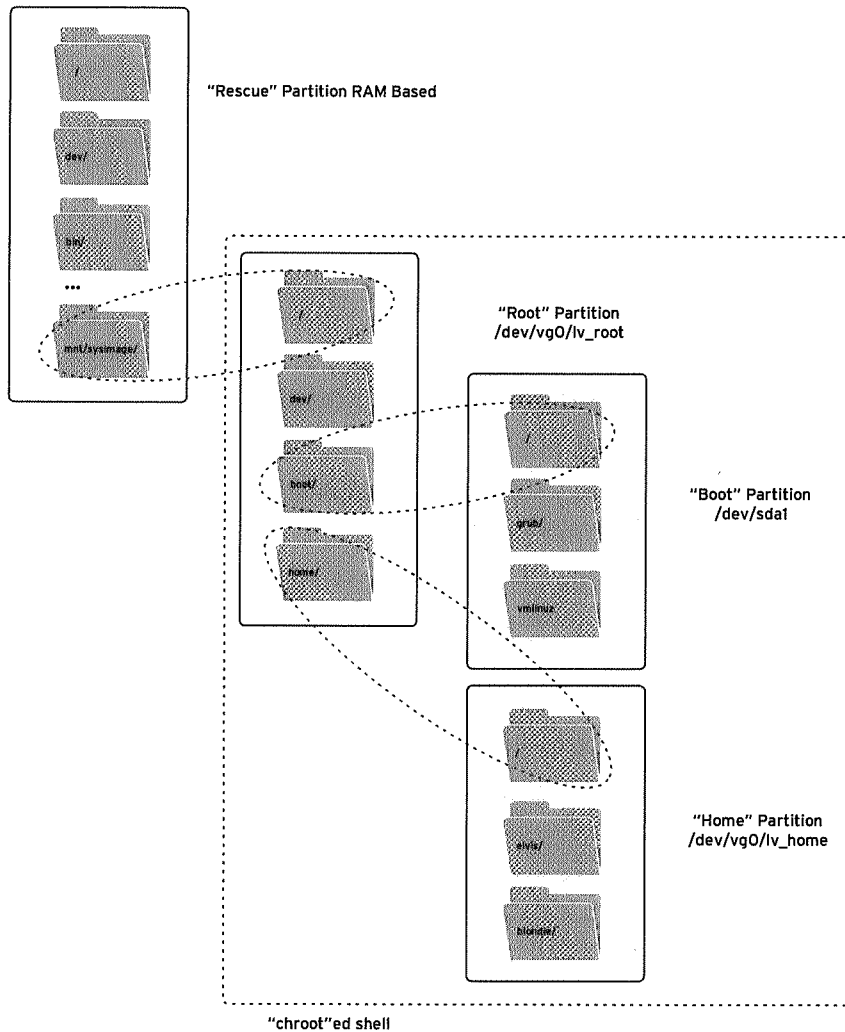
Note

Red Hat Enterprise Linux 6 supports systems that use UEFI and the UEFI Boot Manager to load the operating system instead of a BIOS and GRUB. For more information, see the *Red Hat Enterprise Linux Installation Guide*.

The Rescue Shell

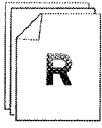
Recovery modes are useful when you are able to use GRUB, but what if GRUB is broken? The Rescue shell, a special mode of the installer, allows you to boot and recover the system when it is otherwise unbootable.

To enter the rescue shell, start as you would an installation, but choose *Rescue Installed System* from the initial menu or append **rescue** as an argument to the kernel.



The first thing you often want to do in the rescue shell is to access or recover filesystems on your local hard drive. Rescue mode attempts to mount your system's root filesystem (and others) under `/mnt/sysimage`, as pictured above.

A useful trick is to use **chroot /mnt/sysimage** to start a sub-shell where `/` is your hard drive's root filesystem.



References

Red Hat Enterprise Linux Deployment Guide

- Section 3.2.2: Using RPM - Installing and Upgrading

Red Hat Enterprise Linux Deployment Guide

- Section 23.6: Verifying the Boot Loader

Red Hat Enterprise Linux Installation Guide

- Section 36.1: Rescue Mode

Red Hat Enterprise Linux Installation Guide

- Technical Appendix E: The GRUB Boot Loader

Red Hat Enterprise Linux Installation Guide

- Technical Appendix F: Boot Process, Init, and Shutdown

info chroot

info grub



Practice Exercise

Using the Rescue Environment

Before you begin...

Run **lab-setup-bootbreak** on desktopX.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

This timed drill is designed to give you practice accessing the rescue environment. The installation path is <http://instructor.example.com/pub/rhel6/dvd>. The path for individual packages is <http://instructor.example.com/pub/rhel6/dvd/Packages/>

1. After serverX has booted, run the **lab-setup-bootbreak-0** script. This script will alter you system and cause difficulties booting.
2. Boot into the rescue environment to diagnose and resolve the issue.
3. Confirm the problem has been solved by rebooting the system.
4. Repeat this process as often as possible during the allotted time.

Repairing Boot Issues

Reinstalling GRUB

First stage GRUB is a small binary that is installed into the Master Boot Record (MBR) of a bootable disk. Usually, GRUB is installed by the Anaconda installer, and never has to be re-installed. Occasionally, if a disk is damaged or moved, you may need to reinstall GRUB manually.

Procedure for Reinstalling GRUB

1. Invoke GRUB

```
[root@serverX ~]# grub
```

2. Identify the **/boot** partitioning

```
grub> root (hd0,0)
```

GRUB refers to hard drives as **(hd0)** or **(hd1)**, which refers to "BIOS drive #0" or "BIOS drive #1". The actual drive can vary depending on BIOS. The first partition on **(hd0)** would be **(hd0,0)**, which might be **/dev/sda1** or **/dev/vda1**.

3. Install first stage grub into the MBR

```
grub> setup (hd0)
```

4. Exit grub

```
grub> quit
```

Repairing Damaged Filesystems

In normal operation, the kernel keeps frequently accessed filesystem information in memory, and only periodically commits the information to disk. If a filesystem becomes unexpectedly unavailable (such as due to a power outage or physical connectivity problem), the on-disk filesystem will contain inconsistencies. If these errors are not fixed, they are likely to lead to corrupt data.

The **fsck** command will attempt to restore a filesystem to a self-consistent state. The guarantee of **fsck** is not full data recovery, but consistency of the filesystem. On boot, the startup scripts will automatically **fsck** all filesystems (if tagged in **/etc/fstab**). Minor problems will be resolved without interaction. If an automatic repair to a filesystem risks data loss, the boot process will drop to a shell to allow an administrator to run **fsck** interactively. If the root partition is damaged, it might be necessary to boot the rescue environment to run **fsck**.

1. Unmount the **/boot** filesystem on **/dev/vda1**.

```
[root@demo ~]# umount /dev/vda1
```

2. Check the filesystem on **/dev/vda1**.

```
[root@demo ~]# fsck /dev/vda1
```

3. Remount the **/boot** filesystem.

```
[root@demo ~]# mount /dev/vda1
```

Procedure for Editing Files from the Maintenance Shell

This is sometimes necessary when a system can not mount filesystems due to typos in **/etc/fstab**, **/etc/crypttab**, or related files, and the system drops to a maintenance shell with / mounted read-only.

1. Remount the root file-system read-write:

```
(Repair filesystem 1)# mount -o remount,rw /
```

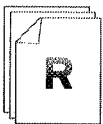
2. Mount all other filesystems (if needed):

```
(Repair filesystem 2)# mount -a
```

3. Edit any needed files.

4. Exit the maintenance shell:

```
(Repair filesystem 4)# exit
```



References

Red Hat Magazine: "Using GRUB to overcome boot problems"

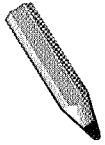
<http://magazine.redhat.com/2007/03/21/using-grub-to-overcome-boot-problems/>



Practice Quiz

Troubleshooting Quiz

1. In maintenance mode, run _____ to mark the / partition as writable.
2. Assuming you have only one hard drive, and the first partition contains /boot, if you have minor MBR corruption, fix the corruption issue by booting into _____ mode, then run the _____ command. Type _____ followed by _____, then exit.
3. If you have filesystem corruption issues, the machine will boot into _____ mode.
4. In maintenance mode, run _____ to fix _____ corruption issues.



Test

Criterion Test

Exercise

Troubleshooting the Boot Process

Before you begin...

Run the **lab-setup-bootbreak** command on desktopX to setup the lab.

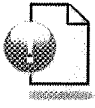
Carefully perform the following steps. Ask your instructor if you have problems or questions.

This practice exercise includes three break/fix challenges. For each, your serverX virtual machine will be modified in some way that prevents it from booting correctly and you will diagnose and correct the problem.

1. Run **lab-setup-bootbreak-1** on serverX. After running the script, serverX should no longer boot correctly. Diagnose and correct the problem. You will know you have the solution when serverX boots normally again.
2. When you have resolved the first scenario, repeat the process with **lab-setup-bootbreak-2** and **lab-setup-bootbreak-3**.



Personal Notes



Unit Summary

Resolve GRUB Issues

In this section you learned how to:

- Use GRUB to correct a broken GRUB configuration so the system will boot

Making Persistent GRUB Changes

In this section you learned how to:

- Persistently correct a GRUB misconfiguration
- Configure the system to boot from a different default kernel

Changing the Default Run Level

In this section you learned how to:

- Use GRUB to boot the system into a specific runlevel

Single-User Mode

In this section you learned how to:

- Enter single-user mode
- Use single-user mode to repair boot problems

The Boot Process and Rescue Mode

In this section you learned how to:

- Describe the boot process for Red Hat Enterprise Linux 6

Repairing Boot Issues

In this section you learned how to:

- Reinstall GRUB
- Check and repair errors on file systems
- Edit files from the read-only file system maintenance shell



UNIT NINE

SELINUX MANAGEMENT

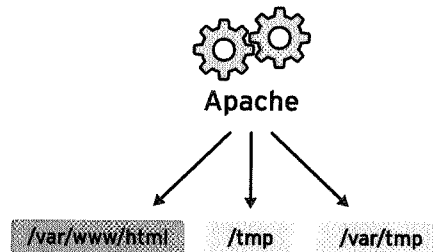
Introduction

Topics covered in this unit:

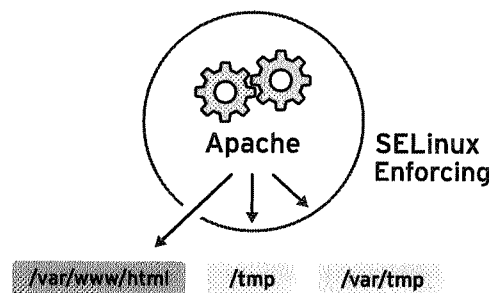
- Review basic SELinux concepts
- Displaying and setting SELinux modes
- Displaying and setting SELinux file contexts
- Tuning policy behavior with SELinux booleans
- Monitoring SELinux policy violations

Basic SELinux Security Concepts

SELinux, Security-Enhanced Linux, is an additional method to protect your system.



Presuming we want to allow remote anonymous access to a web server, we must open the ports through the firewall. However, that means that malicious people can try to crack into the system through a security exploit and, if they compromise the web server process, gain its permissions: the permissions of the apache user and the apache group. That user/group has read access to things like the document root (`/var/www/html`), as well as write access to `/tmp`, `/var/tmp` and any other files/directories that are world writable.

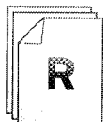


SELinux is a set of security rules that determine which process can access which files, directories, ports, etc. Every file, process, directory and port have special security label called SELinux contexts. A context is simply a name that is used by the SELinux policy to determine whether or not a process can access a file, directory or port. By default, the policy does not allow any interaction, so explicit rules grant access. If there is no allow rule, no access is allowed.

SELinux labels have several contexts, but we are most interested in the third context: the type context. Type context names usually end with `_t`. The type context for the web server is `httpd_t`. The type context for files and directories normally found in `/var/www/html` is `httpd_sys_content_t`. The type contexts for files and directories normally found in `/tmp` and `/var/tmp` is `tmp_t`. The type context for web server ports is `http_port_t`.

There is a rule in the policy that permits Apache (the web server process running as `httpd_t`) to access files and directories with a context normally found in `/var/www/html` and other web server directories (`httpd_sys_content_t`). There is no allow rule in the policy for files normally found in `/tmp` and `/var/tmp`, so access is not permitted. With SELinux, a malicious user could not access the `/tmp` directory, let alone write files to it. SELinux even has rules for remote filesystems such as NFS and CIFS, although all files on these filesystems are labeled with the same context.

One of the goals of SELinux is to protect the user data from system services that have been compromised.



References

Red Hat Enterprise Linux SELinux Guide

- Section 2: Introduction



Practice Quiz

Basic SELinux Concepts

1. To which of the following does SELinux apply security context (check all that apply)?

(select one or more of the following...)

- a. Ports
- b. Processes
- c. Files
- d. Directories
- e. Remote file systems

2. SELinux can be used to:

(select one or more of the following...)

- a. Protect a service from running on other ports.
- b. Protect user data from applications like the web server
- c. Block remote systems from accessing local ports
- d. Keep the system updated
- e. Access a web server

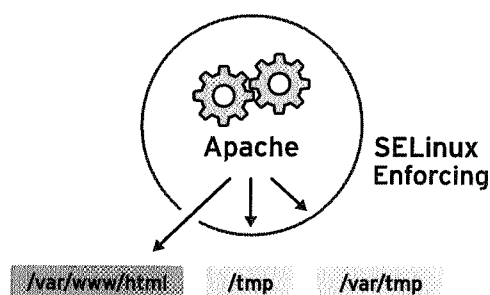
3. Which of the following are standard SELinux context types?

(select one or more of the following...)

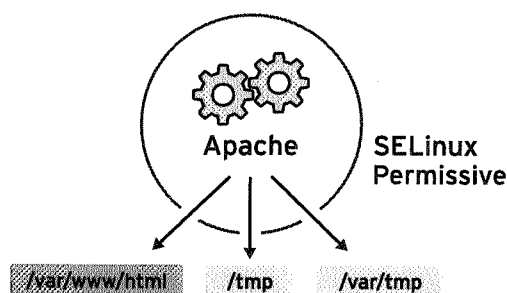
- a. selinux_type
- b. object_r
- c. httpd_sys_content_t
- d. tmp_t
- e. user_u

SELinux Modes

For troubleshooting purposes, we can temporarily disable SELinux protection, using SELinux modes.

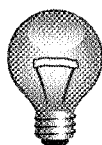


In *enforcing mode*, SELinux actively denies access to the web server attempting to read files with **tmp_t** type context. In enforcing mode, SELinux both logs and protects.



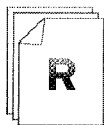
Permissive mode is often used to troubleshoot issues. In permissive mode, SELinux allows all interactions, even if there is no explicit rule, and it logs all of the denied interactions. This mode can be used to determine if you are having an SELinux issue. No reboot is required to go from enforcing to permissive or back again.

A third mode, *disabled*, completely disables SELinux. You must reboot to disable SELinux entirely, or to get from disabled mode to enforcing or permissive.



Important

If you plan to re-enable SELinux restrictions, it is better to use permissive mode than to turn off SELinux entirely. One reason for this is that even in permissive mode, the kernel will automatically maintain SELinux file system labels as needed, avoiding the need for an expensive relabeling of the file system when you reboot with SELinux re-enabled.



References

Red Hat Enterprise Linux SELinux Guide

- Section 5.5: SELinux Modes



Practice Quiz

SELinux Modes

1. SELinux _____ mode allows logging, but not protection.
2. SELinux _____ mode protects the system.
3. Which of the following are valid SELinux modes?
(select one or more of the following...)
 - a. enforcing
 - b. testing
 - c. permissive
 - d. disabled
 - e. logging

Display and Modify SELinux Modes

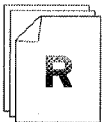
Notice that `/etc/sysconfig/selinux` contains some useful comments:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Use `/etc/sysconfig/selinux` to change the default SELinux mode at boot time. In the example above, it is set to enforcing mode.

To display the current SELinux mode, use **getenforce**. To modify the current SELinux mode, use **setenforce**:

```
[root@serverX ~]# getenforce
Enforcing
[root@serverX ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@serverX ~]# setenforce 0
[root@serverX ~]# getenforce
Permissive
[root@serverX ~]# setenforce Enforcing
[root@serverX ~]# getenforce
Enforcing
```



References

Red Hat Enterprise Linux SELinux Guide
• Section 5.5: SELinux Modes

selinux(8), **getenforce(1)**, **setenforce(1)** man pages

**Practice Exercise****Changing Enforcing and Permissive Modes**

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. On serverX, change the default SELinux mode to permissive and reboot.
2. After reboot, verify the system is in permissive mode.
3. Change the default SELinux mode to enforcing.
4. Change the current SELinux mode to enforcing.

Display and Modify SELinux File Contexts

Many commands that deal with files have an option (usually **-Z**) to display or set SELinux contexts. For instance, **ps**, **ls**, **cp**, and **mkdir** all use the **-Z** option to display or set SELinux contexts.

```
[root@serverX ~]# ps axZ
LABEL                                PID TTY          STAT TIME COMMAND
system_u:system_r:init_t:s0          1 ?        Ss   0:00 /sbin/init
system_u:system_r:kernel_t:s0        2 ?        S    0:00 [kthreadd]
system_u:system_r:kernel_t:s0        3 ?        S    0:00 [migration/0]
...
[root@serverX ~]# service httpd start
[root@serverX ~]# ps -ZC httpd
LABEL                                PID TTY          TIME CMD
unconfined_u:system_r:httpd_t:s0     27672 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0     27675 ?        00:00:00 httpd
...
[root@serverX ~]# ls -Z /home
drwx-----. root root system_u:object_r:lost_found_t:s0 lost+found
drwx-----. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@serverX ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

What determines a file's initial SELinux context? Normally, it is the parent directory. The context of the parent directory is assigned to the newly-created file. This works for commands like **vim**, **cp**, and **touch**, however, if a file is created elsewhere and the permissions are preserved (as with **mv** or **cp -a**), it will preserve the SELinux context as well. There are some special rules in the policy, called type transition rules, that may change the type context from the default. These rules are beyond the scope of this course.

```
[root@serverX ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
[root@serverX ~]# touch /var/www/html/index.html
[root@serverX ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

semanage fcontext can be used to display or modify the rules that **restorecon** uses to set default file contexts. It uses extended regular expressions to specify the path and file names. The most common extended regular expression used in **fcontext** rules is **(/.*)?** which means *optionally, match a / followed by any number of characters*. In essence, it will match the directory listed before the expression and everything in that directory recursively.

restorecon is part of the **policycoreutil** package, and **semanage** is part of the **policycoreutil-python** package.

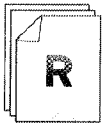
```
[root@serverX ~]# touch /tmp/file1 /tmp/file2
[root@serverX ~]# ls -Z /tmp/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```



```
[root@serverX ~]# mv /tmp/file1 /var/www/html/
[root@serverX ~]# cp /tmp/file2 /var/www/html/
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
[root@serverX ~]# semanage fcontext -l
...
/var/www(/.*)?                                all files
system_u:object_r:httpd_sys_content_t:s0
...
[root@serverX ~]# restorecon -Rv /var/www/
restorecon reset /var/www/html/file1 context unconfined_u:object_r:user_tmp_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
[root@serverX ~]# ls -Z /var/www/html/file*
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

The following example show how to use **semanage** to add a context for a new directory.

```
[root@serverX ~]# mkdir /virtual
[root@serverX ~]# touch /virtual/index.html
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html
[root@serverX ~]# semanage fcontext -a -f "" -t httpd_sys_content_t '/virtual(/.*)?'
[root@serverX ~]# restorecon -RFvv /virtual
[root@serverX ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@serverX ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```



References

Red Hat Enterprise Linux SELinux Guide
 • Section 5.7: SELinux Contexts - Labeling Files

restorecon(8) and **semanage**(8) man pages



Practice Exercise

Correcting SELinux file contexts

Carefully perform the following steps. Ask your instructor if you have problems or questions.

You have been asked to adjust your remote machine's DNS configuration to exactly match the configuration from your desktop machine. You decide the easiest way is to copy the file **/etc/resolv.conf** from the local machine to the remote machine.

1. Transfer the **/etc/resolv.conf** file from your desktop machine to **root**'s home directory on serverX.
2. Shell into serverX as **root**. All of the following steps should occur on your server.
3. Observe the SELinux context of the initial **/etc/resolv.conf**.

Original **/etc/resolv.conf** context:

4. Move **resolv.conf** from **root**'s home directory to **/etc/resolv.conf**.
5. Observe the SELinux context of the newly copied **/etc/resolv.conf**.

New **/etc/resolv.conf** context:

6. Restore the SELinux context of newly positioned **/etc/resolv.conf**.
7. Observe the SELinux context of the restored **/etc/resolv.conf**.

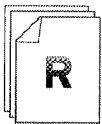
Restored **/etc/resolv.conf** context:

Managing SELinux Booleans

SELinux booleans are switches that change the behavior of the SELinux policy. SELinux booleans are rules that can be enabled or disabled. They can be used by security administrators to tune the policy to make selective adjustments. Many packages have man pages ***_selinux(8)** which may detail some of the booleans which they use; **man -k '_selinux'** can find these man pages easily.

getsebool is used to display the booleans and **setsebool** is used to modify the booleans. **setsebool -P** modifies the SELinux policy to make the modification persistent. **semanage boolean -l** will show whether or not a boolean is persistent.

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
...
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
[root@serverX ~]# setsebool httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          -> off    Allow httpd to read home directories
[root@serverX ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
[root@serverX ~]# setsebool -P httpd_enable_homedirs on
[root@serverX ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          -> on     Allow httpd to read home directories
```



References

Red Hat Enterprise Linux SELinux Guide
 • Section 5.6: Booleans

booleans(8), **getsebool(8)**, **setsebool(8)**, **semanage(8)** man pages



Practice Quiz

SELinux Booleans

What command lists the current state of all SELinux booleans?

What command would enable the *httpd_enable_cgi* boolean immediately?

What command would enable the *ftp_home_dir* boolean immediately, and across reboots?

What command displays current SELinux boolean configuration, as well as short annotations about the boolean?

Monitor SELinux Violations

The **setroubleshoot-server** package must be installed to send SELinux messages to **/var/log/messages**. **setroubleshoot-server** listens for audit messages in **/var/log/audit/audit.log** and sends a short summary to **/var/log/messages**. This summary includes unique identifiers (**UUIDs**) for SELinux violations that can be used to gather further information. **sealert -l UUID** is used to produce a report for a specific incident. **sealert -a /var/log/audit/audit.log** is used to produce reports for all incidents in that file.

```
[root@serverX ~]# touch /root/file3
[root@serverX ~]# mv /root/file3 /var/www/html
[root@serverX ~]# service httpd start
[root@serverX ~]# elinks -dump http://serverX/file3
Forbidden
You don't have permission to access /file3 on this server.
[root@serverX ~]# tail /var/log/audit/audit.log
...
type=AVC msg=audit(1292526292.144:952): avc: denied { getattr } for
pid=27675 comm="httpd" path="/var/www/html/file3" dev=dm-1 ino=54545
scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:admin_home_t:s0
tclass=file
...
[root@serverX ~]# tail /var/log/messages
...
Dec 16 14:04:59 serverX setroubleshoot: SELinux is preventing /usr/sbin/httpd "getattr"
access to /var/www/html/file3. For complete SELinux messages. run sealert -l e6e1d1d6-
d716-4e2e-863c-bba4d2b2407a
[root@serverX ~]# sealert -l e6e1d1d6-d716-4e2e-863c-bba4d2b2407a
Summary:

SELinux is preventing /usr/sbin/httpd "getattr" access to /var/www/html/file3.

Detailed Description:

SELinux denied access requested by httpd. /var/www/html/file3 may be a
mislabelled. /var/www/html/file3 default SELinux type is httpd_sys_content_t, but
its current type is admin_home_t. Changing this file back to the default type,
may fix your problem.
...
Allowing Access:

You can restore the default system context to this file by executing the
restorecon command. restorecon '/var/www/html/file3', if this file is a
directory, you can recursively restore using restorecon -R
'/var/www/html/file3'.

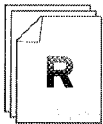
Fix Command:

/sbin/restorecon '/var/www/html/file3'
...
```



Note

The "Allowing Access" section suggests **restorecon /var/www/html/file3**. If there may be other files that need to be adjusted, **restorecon** can recursively reset the context: **restorecon -R /var/www/**.



References

Red Hat Enterprise Linux SELinux Guide

- Chapter 8: Troubleshooting

Red Hat Enterprise Linux SELinux Guide

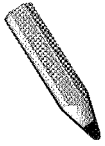
- Section 8.3.7: **sealert** Messages

sealert(8) man page

**Practice Quiz****Monitoring SELinux Violations**

1. What file contains log entries providing unique identifiers for SELinux violations?

2. Given the UUID of an SELinux violation, what command generates a text report of the problem?



Test

Criterion Test

Exercise

Managing SELinux

Before you begin...

Before you begin, run the **lab-setup-selinux** command on desktopX

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Login to serverX as **student**. Open a terminal and switch to the **root** user.
2. Copy the *web_content.tgz* archive from *instructor:/var/ftp/pub/materials* to **/tmp**.
3. Extract the archive into **/tmp**.
4. Move the extracted directory to **/var/www/html**.
5. Start the web service.
6. Try to observe the new directory with your web browser by visiting the URL *http://serverX/web_content*.
7. Search your system for the UUIDs of any SELinux violations your attempt to browse the newly installed content might have generated.
8. Generate text reports for the violations.
9. Follow the report's advice to restore the SELinux contexts of the newly installed content.
10. Confirm that you can view the material from your web browser by visiting the URL *http://serverX/web_content*.



Personal Notes



Unit Summary

Basic SELinux Security Concepts

In this section you learned how to:

- Identify basic SELinux security concepts such as context, user/role/type, and policy

SELinux Modes

In this section you learned how to:

- Describe the functional differences between SELinux enforcing and permissive modes when SELinux security is enabled

Display and Modify SELinux Modes

In this section you learned how to:

- View and change the current SELinux mode of a system
- Set the default SELinux mode of a system

Display and Modify SELinux File Contexts

In this section you learned how to:

- View the SELinux security context of processes and files
- Set the SELinux security context of files in the policy
- Restore the SELinux security context of files

Managing SELinux Booleans

In this section you learned how to:

- Use SELinux booleans to make adjustments to policy behavior

Monitor SELinux Violations

In this section you learned how to:

- Deploy SELinux log analysis tools



UNIT TEN

FIREWALL MANAGEMENT

Introduction

Topics covered in this unit:

- Packet filtering
- Network address translation (NAT)

Packet Filtering

The following is a list of key concepts you will need to know in order to setup a firewall. Pay attention as the class discusses each one and take notes in your books because you will have to use all of these keywords when creating your firewall in lab.

- _____ - criteria determining which packets to match and a target, or action, determining what to do with those packets.
- _____ - a list of *rules* which will be checked in order, first match takes effect.
- _____ - the default action, **ACCEPT** or **DROP**, taken if no *rule* matches in a built-in *chain*.
- _____ - a set of *chains* used for a particular purpose: **filter** to block traffic, **nat** to modify the destination or apparent source of a packet.

Built-in Chains (filter table)

- _____ - packets addressed to the firewall
- _____ - packets originating from a service on the firewall (not forwarded)
- _____ - packets originating from another machine, that are not addressed to the firewall but are being forwarded (routed) elsewhere (when `net.ipv4.ip_forward=1`)

Targets

(Actions to take when packets match rules)

- _____ - the packet passes the chain
- _____ - the packet is dropped as if it was never seen
- _____ - the packet is rejected, and the firewall sends an error message (an ICMP port unreachable message by default)
- _____ - information about the packet is logged to syslog; we go on to the next rule in the chain

iptables Command

You may have used **system-config-firewall**, a graphical tool in Red Hat Enterprise Linux 6, to configure simple firewalls. Creating and managing more advanced configurations can be accomplished with the command line tool, **iptables**.

iptables is used to set or view rules in kernel memory.

<i>iptables Options</i>	<i>Definition</i>
-vnL --line-numbers	lists all rules, fully, in numeric mode
-A CHAIN <rule> -j <target>	adds a <i>rule</i> to the end of <i>CHAIN</i>
-I CHAIN # <rule> -j <target>	inserts a <i>rule</i> as rule # in <i>CHAIN</i> ; if no #, then as the first rule
-D CHAIN #	deletes rule # from <i>CHAIN</i>
-F CHAIN	deletes all rules from <i>CHAIN</i>

Table 10.1. iptables Example Syntax

Rule (matching criteria) Syntax

An iptables rule includes matching criteria that can compares to header information found in the packet.

<i>Concept</i>	<i>Directive</i>
Source IP or network	-s 192.0.2.0/24
Destination IP or network	-d 10.0.0.1
UDP/TCP and ports	-p udp --sport 68 --dport 67
ICMP and types	-p icmp --icmp-type echo-reply
Inbound network interface	-i eth0
Outbound network interface	-o eth0
State tracking	-m state --state ESTABLISHED,RELATED

Table 10.2. iptables Matching Criteria

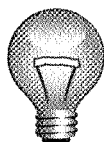
State tracking stores information about previously seen communications to make matching decisions. After connection allowed, information placed in a state tracking table until a timeout occurs, connections closes, or we see more matching traffic (reset timer). While this takes additional kernel memory, the benefit is to simplify rule design.

<i>State</i>	<i>Definition</i>
NEW	packet starts a new communication, adds a rule to the state tracking table
ESTABLISHED	any packet that matches a rule in the state tracking table
RELATED	traffic "related" in some way to ESTABLISHED traffic; protocols like FTP

State	Definition
INVALID	packet cannot be identified; normally these should be rejected or dropped

Table 10.3. Connection Tracking States

To help RELATED rules work, you may need to enable helper modules in **/etc/sysconfig/iptables-config**

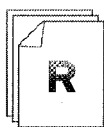


Important

Running the **iptables** command changes the netfilter kernel module rules in memory, but will NOT persist across a reboot.

Running **service iptables save** will take the current rules in memory and write them to **/etc/sysconfig/iptables** which is read during startup.

Alternatively, some administrators directly edit (or copy) the **/etc/sysconfig/iptables** file, then run **service iptables restart** to activate.



References

Red Hat Enterprise Linux Security Guide

- Section 2.5: Firewalls

Red Hat Enterprise Linux Security Guide

- Section 2.6: IPTables

iptables(8) man page

Netfilter home page

<http://www.netfilter.org/>



Practice Exercise

Implement a Firewall

Carefully perform the following steps. Ask your instructor if you have problems or questions.

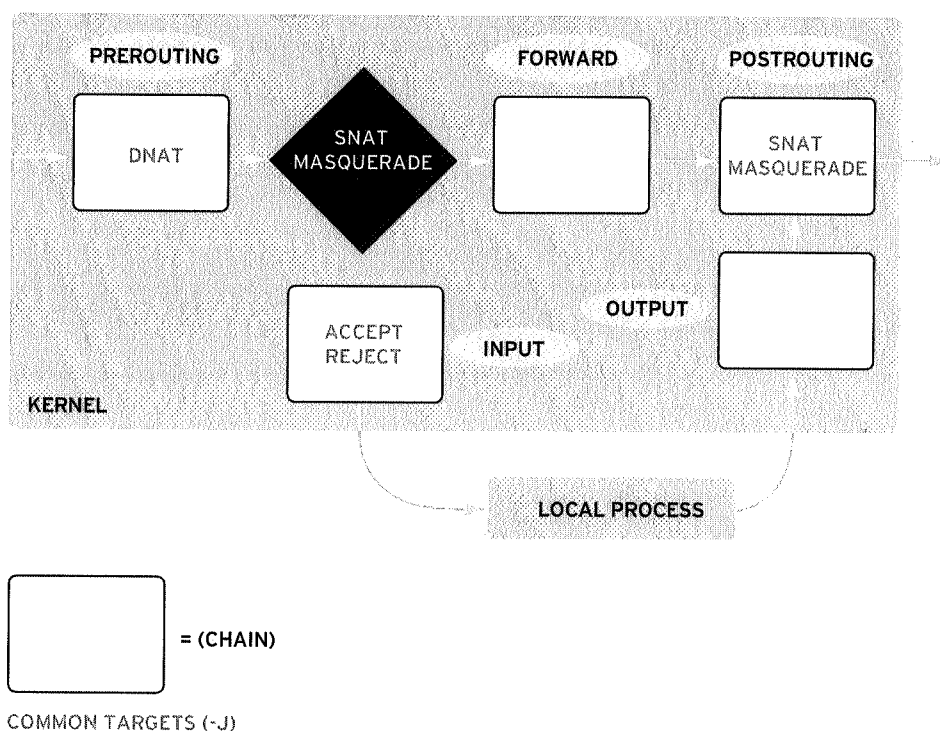
In this exercise you will implement a firewall on serverX that rejects all packets, except that it will allow ICMP traffic for example.com and allow SSH for everyone except remote.test.

1. Log into serverX as **root** using **virt-viewer** or **virt-manager**.
2. Create a simple deny all (except loop back) firewall by creating **/root/bin/resetfw.sh** that
 1. sets the **INPUT** chain's default policy to **DROP**,
 2. flushes all rules in the filter table, and
 3. will **ACCEPT** all packets from the loopback interface
3. Run your script and record the results of the following:
 - **ping** and **ssh** serverX from desktopX and from remoteX.remote.test
4. What happens when you **ping** desktopX and 192.168.0.X from serverX now? Why?
5. Enable stateful firewalling by appending to your script a rule that will
 - **ACCEPT** all **ESTABLISHED, RELATED** packets
6. Run your script and record the results of the following:
 - **ping** desktopX and 192.168.0.X from serverX
7. Reject all packets from remote.test by appending to your script a rule that will
 - **REJECT** all packets from the 192.168.1.0/24 network
8. Run your script and record the results of the following:
 - **ping** and **ssh** serverX from desktopX and from remoteX.remote.test
9. Enable ICMP traffic for example.com by appending to your script a rule that will
 - **ACCEPT** all **icmp** traffic from 192.168.0.0/24
10. Run your script and record the results of the following:
 - **ping** and **ssh** serverX from desktopX
11. Enable SSH traffic for all hosts by modifying your script to
 - **ACCEPT** all **NEW** connections to **tcp** port **22**
12. Run your script and record the results of the following:

- **ssh** to serverX from desktopX and from remoteX.remote.test
13. Reject packets by default instead of dropping packets by appending to your script a rule that will
- **REJECT** all other traffic
14. Run your script and record the results of the following:
- **ping** and **ssh** serverX from desktopX and from remoteX.remote.test

Network Address Translation

Network Address Translation is used to manipulate the apparent source or desired destination address of network packets. The diagram below shows the order in which Netfilter chains on the **filter** and **nat** tables are processed:



Simple host-based firewalls may only have rules in the **INPUT** chain to **ACCEPT** or **REJECT** packets, but on a gateway or router for a private (non-routable) network, it is common to use the **PREROUTING** and **POSTROUTING** chain to modify packets.

The **nat** table uses three chains: **PREROUTING**, **OUTPUT**, and **POSTROUTING**. Network Address Translation is when a router modifies the source or destination IP address or port of network traffic passing through it. It is used for mapping a network of machines behind a single IP address, so that they may share a single public address and hide their internal network (**MASQUERADE** or **SNAT**). It is also used for redirecting traffic addressed to one IP address to another. This *destination NAT* is used for *port forwarding* (passing a port outside a firewall to a service inside it) and for transparent redirection to proxy services.

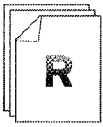
The **MASQUERADE** target causes the source IP address to be changed to match the IP of the interface it leaves the firewall on. Destination sends response back to the IP address of that interface. State tracking automatically de-masquerades the IP address to the right original source (tracks based on IP addresses and ports of both ends of the connection). The **SNAT** target causes the source IP address to change to a specified IP address with the option **--to-source**.

```
[root@demo ~]# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

The **DNAT** target causes the destination IP address to be changed to match the IP address specified by the **--to-destination** option. Router forwards packet to that address; this is why this chain is before the routing decision. State tracking automatically sends responses back to the original source with the original IP address on it, not the new one.

```
[root@demo ~]# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.0.254
```

Use this space for notes



References

Red Hat Enterprise Linux Security Guide

- Section 2.5: Firewalls

Red Hat Enterprise Linux Security Guide

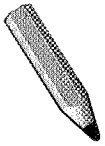
- Section 2.6: IPTables



Practice Quiz

Network Address Translation Quiz

1. The chains available in the **filter** table are _____, _____, and _____.
2. The chains available in the **nat** table are _____, _____, and _____.
3. **iptables -t _____ -A _____ -o eth0 -j MASQUERADE**
4. **iptables -t _____ -A _____ -o eth0 -j SNAT _____ 192.168.0.1**
5. **iptables -t _____ -A _____ -i eth0 -m tcp -p tcp --dport 80 -j DNAT _____ 192.168.0.100:8080**
6. The **DNAT** target can only be used in the _____ chain and the _____ chain of the _____ table
7. To enable forwarding persistently across reboots add **net _____ =1** to **/etc/_____** and run **_____ -p**



Test

Criterion Test

Case Study

The Morris Worm and Fish Supply Company

Before you begin...

Important: Make sure to run the **lab-setup-morrisworm** script on desktopX before you begin! The **lab-setup-morrisworm** script will configure serverX to run on a private network.

The Morris Worm and Fish Supply company is finally looking to modernize its business by opening a website. The web server will run on a private network behind a firewall. The firewall will forward all TCP port 80 traffic to the web server and will perform NAT so that the web server can reach external hosts.

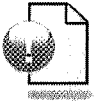
- desktopX.example.com will be the firewall, serverX.example.com will be the web server.
- Configure Apache to run on serverX.example.com. Put some custom content in **/var/www/html/index.html** that will uniquely identify the server.
- Configure the firewall on desktopX to perform NAT that will allow the web server to reach the outside network. You will be able to successfully **ping** instructor.example.com from serverX to confirm this works.
- Finally configure the firewall to forward all TCP port 80 traffic sent to it to the web server running on serverX. You will need to identify serverX's IP address to complete this step. Confirm this works by using a web browser from an external machine, NOT desktopX, to browse <http://desktopX.example.com>.

After you have successfully completed the lab, run **lab-cleanup-morrisworm** on desktopX to reset your network back to its original state.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Packet Filtering

In this section you learned how to:

- Set firewall rules with **iptables**
- Block or allow network traffic based on specific criteria
- Block or allow network traffic based on previous traffic seen

Network Address Translation

In this section you learned how to:

- Use **iptables** to set up IPv4 Network Address Translation
- Make packets passing through the Linux router appear to come from its outbound IP address
- Redirect packets passing through the Linux router to a different destination IP address



UNIT ELEVEN

NTP SERVER CONFIGURATION

Introduction

Topics covered in this unit:

- Configuring time servers

Configure an NTP Server

NTP is the Network Time Protocol, a standard way for machines to provide and obtain correct time information on the Internet. A machine may get accurate time information from public NTP services on the Internet such as the NTP Pool Project, or from high-quality hardware clocks, which it then may itself serve to local clients.

To configure an NTP server and client, you need to understand three main parameters in the `/etc/ntp.conf` file: **server**, **peer**, and **restrict**.

The first argument of the **server** line is the IP address or DNS name of the NTP server. Your IP address must be allowed access on the server with a **restrict** line discussed below. Following the server IP address or name, you can list a series of options for the server. The `ntp.conf(5)` man page recommends using the **iburst** option.

Like the **server** line, the **peer** line takes an NTP server and options arguments. The **server** is one stratum above your NTP server, and the **peer** is at the same stratum. You can specify more than one **server** and more than one **peer**, one per line.

The **restrict** line usually takes an IP address or DNS name as the first argument. However, the first argument could be **default** to specify the restrictions apply as default settings. If the first argument is **-6**, the following restrictions (including **default**) apply to IPv6 addresses only. The **restrict** line has several flags that can be used; some of the most common are explained below (taken from the `ntp_acc(5)` man page:

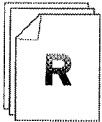
- **ignore**: Deny packets of all kinds, including **ntpq** and **ntpd** queries
- **kod**: If this flag is set when an access violation occurs, a kiss-o'-death (KoD) packet is sent. KoD packets are rate limited to no more than one per second. If another KoD packet occurs within one second after the last one, the packet is dropped
- **nomodify**: Deny **ntpq** and **ntpd** queries which attempt to modify the state of the server (i.e., run time reconfiguration). Queries which return information are permitted.
- **noquery**: Deny **ntpq** and **ntpd** queries. Time service is not affected.
- **nopeer**: Deny packets which would result in mobilizing a new association. This includes broadcast, symmetric-active and manycast client packets when a configured association does not exist.
- **notrap**: Decline to provide mode 6 control message trap service to matching hosts. The trap service is a subsystem of the **ntpd** control message protocol which is intended for use by remote event logging programs.

If the client time is off by more than just a few minutes, the NTP client will fail to sync with the server. You can use the `ntpdate -v ntpserver` command to roughly set the clock once so that it can synchronize.

If you want to include a hardware clock available to the machine as part of the NTP service, it uses a special IP address in the range **127.127.1**. This could be an accurate GPS or radio clock, or the inaccurate RTC built into the system.

Example: The real-time clock (RTC) is used to keep track of time on the motherboard. This clock is usually not very accurate. For this reason, if you use it you should force it to advertize at a lower stratum (normally 10). The section in **/etc/ntp.conf** would look like the following:

```
server      127.127.1.1
fudge       127.127.1.0 stratum 10
```



References

Red Hat Enterprise Linux Deployment Guide

- Section 13.2.2: Network Time Protocol Setup

ntp.conf(5) and **ntp_acc**(5) man pages

/usr/share/doc/ntp-*/html (from the **ntp-doc** package)

NTP Pool Project
<http://www.ntp.pool.org/>



Practice Quiz

Configuring NTP Quiz

Answer the questions below based upon the following NTP configuration file:

```
#/etc/ntp.conf

restrict default kod nomodify notrap nopeer noquery
restrict -6 default ignore

restrict 192.168.0.0 mask 255.255.255.0 nomodify notrap nopeer
restrict 192.168.0.101 kod nomodify notrap
restrict 192.168.0.200

server 192.168.0.2
server 192.168.0.3
peer 192.168.0.101
```

1. The NTP client's time is off by 15 minutes, it will eventually sync with the servers.
(select one of the following...)
 - a. True
 - b. False

2. The NTP client will use the computer's RTC (BIOS) as a time source.
(select one of the following...)
 - a. True
 - b. False

3. 192.168.0.200 will be able to modify the time on this NTP server.
(select one of the following...)
 - a. True
 - b. False

4. 192.168.0.4 will be able to query this NTP server.
(select one of the following...)
 - a. True
 - b. False

5. 192.168.0.3 will be able to use this NTP server as a peer.
(select one of the following...)
 - a. True
 - b. False

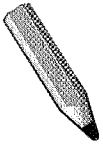
6. Anyone with an IPv4 address will be able use this NTP server as a time source.
(select one of the following...)

- a. True
- b. False

7. Anyone with an IPv6 address will be able use this NTP server as a time source.

(select one of the following...)

- a. True
- b. False



Test

Criterion Test

Case Study

NTP Server Configuration

Before you begin...

Run **lab-setup-howsonclock** on desktopX.

Howson Heavy Machine and Clock Manufacture, maker of clock tower parts and accessories, recently conducted an audit of all computer systems. The audit revealed several systems with out of sync clocks, including your serverX.example.com machine.

Set up NTP on your serverX to be a client of the NTP service running on instructor.example.com.

In order to have additional time sources, work with a few neighbors so that all of your serverX systems are set up to synchronize as NTP peers.

When you have finished, run **lab-grade-howsonclock** on desktopX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Configure an NTP Server

In this section you learned how to:

- Setup cluster of NTP servers as peers and use lower strata service
- Configure clients to use your local cluster of NTP servers



UNIT TWELVE

SYSTEM LOGGING SERVICE

Introduction

Topics covered in this unit:

- System monitoring scripts
- Centralized logging

Usage Reports

In this section we will begin by look at a number of tools that are useful when writing scripts or automated tasks to collect reports on the usage of the system.

Usage Reports Buzz Groups

1. **df** is used to show disk usage. The **-h** option prints the output in "human-readable" form.

```
[root@serverX ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/vgsrv-root
                 3.3G  2.2G  935M   71% /
tmpfs            246M  112K   246M    1% /dev/shm
/dev/vda1        248M   30M   206M   13% /boot
/dev/mapper/vgsrv-home
                 248M   11M   225M    5% /home
```

Use this space for notes

2. Create a disk I/O usage report using **iostat**

What does the **-d** option report?

What does the **-N** option report?

What does the **-k** option report?

What does **iostat** do if passed two numeric arguments (e.g., **iostat 2 10**)?

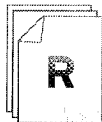
Once you have answered the questions above, turn to the Usage Reports Practice Case Study and follow the instructions there.

3. Create a swap usage report using **vmstat**

What does the first output represent when running **vmstat**?

What does **vmstat** do if passed two numeric arguments (e.g., **vmstat 2 10**)?

Once you have answered the questions above, turn to the Usage Reports Practice Case Study and follow the instructions there.



References

df(1), **iostat(1)**, and **vmstat(8)** man pages



Practice Case Study

Usage Reports

Use the tool you investigated to create a simple report that logs information to a file.

Once you have used the tool to generate a report, the instructor will have you share the command you used and explain the output with the rest of the class.

How would you address the case study described above? Take notes on your process in the space below and then implement it.

Configure a Remote Logging Service

Central collection of system log messages can be very useful for monitoring the state of your systems and for quickly identifying problems. It also provides a backup location for log messages in case a system suffers a catastrophic hard drive failure or other problem where the local logs are no longer available, which can be used to help diagnose the issue that caused the problem.

Standardized system logging is implemented in Red Hat Enterprise Linux 6 by the **rsyslog** service. System programs can send syslog messages to the local **rsyslogd** service, which will then redirect those messages to files in **/var/log**, remote log servers, or other databases based on the settings in its configuration file, **/etc/rsyslog.conf**.

Log messages have two characteristics that are used to sort them; a *facility* that indicates what kind of message it is, and a *priority* that indicates the importance of the event being logged.

Priority	Meaning
emerg	System is unusable
alert	Immediate action required
crit	Critical condition
err	Error condition
warning	Warning condition
notice	Normal but significant condition
info	Informational messages
debug	Debugging messages

Table 12.1. Syslog priority levels

See **logger(1)** and **syslog(3)** for more information and a list of facilities.

Configuring a remote logging service comes in two parts: configuring **rsyslog** on the remote log server to accept log messages from the network, and configuring client **rsyslog** systems to send logs to the remote log server.

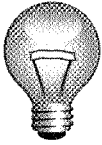
- To configure **rsyslog** to accept remote logs, uncomment either the TCP or UDP reception lines in the modules section in the **/etc/rsyslog.conf** file. For UDP reception:

```
# Provides UDP syslog reception
$ModLoad imudp.so
$UDPServerRun 514
```

or for TCP reception:

```
# Provides TCP syslog reception
$ModLoad imtcp.so
$InputTCPServerRun 514
```

TCP provides more reliable delivery of remote log messages, but UDP is supported by a wider variety of operating systems and networking devices.



Important

Plain TCP transport of syslog messages is fairly widely implemented but not yet standardized. Most implementations currently use port 514/TCP, which is the legacy **rshd** port. If you have installed the *rsh-server* package and are using the old insecure **rshd** service, it will conflict with using port 514/TCP for plain TCP syslog reception. You can configure the log server to use a different port by changing the setting for **\$InputTCPServerRun**.

Once you have uncommented a syslog reception section, restart the **rsyslog** service.

- To configure a machine to send logs to a remote **rsyslog** server, add a line to the rules section in the **/etc/rsyslog.conf** file. In place of the file name, use the IP address of the remote **rsyslog** server. To use UDP, prepend the IP address with a single @ sign. To use TCP, prepend it with two @ signs (@@).

For instance, if you want all messages with **info** or higher priority sent to 192.168.0.1 using UDP, use the following line:

```
*.info @192.168.0.1
```

If you want *all* messages sent to 192.168.0.101 using TCP, use the following line:

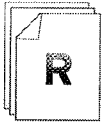
```
*.* @@192.168.0.101
```

Optionally, the IP address can be appended with **:PORT**, where **PORT** is the port that the remote **rsyslog** server is using. If no port is given, it assumes the default port 514.

Once you have added the rule(s), restart the **rsyslog** service and send a test message using the **logger** command:

```
[root@serverX ~]# logger "Test from serverX"
```

Check the logs on the remote server to ensure you received the message.



References

Red Hat Enterprise Linux Deployment Guide

- Chapter 17: Log Files

General **rsyslog** Documentation:

/usr/share/doc/rsyslog-*/manual.html

rsyslog Configuration File Documentation:

/usr/share/doc/rsyslog-*/rsyslog_conf_actions.html

rsyslog.conf(5), **rsyslogd**(8), **logger**(1), **syslog**(3) man pages

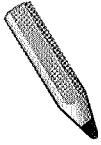


Practice Exercise

Remote Logging

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Configure serverX to accept remote log messages using TCP.
2. Configure desktopX to send all **info** priority and higher events to serverX using TCP .
3. Test your configuration.



Test

Criterion Test

Case Study

System Monitoring and Logs

Before you begin...

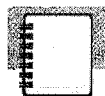
Before you begin, run the **lab-setup-blossoms** script on desktopX.

Blossoms, Inc. is a nation wide cooperative of flower and plant growers. Among other things, the cooperative handles IT services for all members. The IT manager has decided to beef up security by requiring remote logging on all servers, including your serverX.

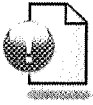
Configure rsyslog on desktopX to accept incoming log messages via UDP from serverX. Then configure **rsyslog** on serverX to send all ***.info** log messages to desktopX via UDP.

When you are ready to check your work, first run **lab-grade-blossoms** on serverX and then run **lab-grade-blossoms** on desktopX.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Usage Reports

In this section you learned how to:

- Verify and monitor filesystem integrity with tools like **df**, **iostat** and **vmstat**

Configure a Remote Logging Service

In this section you learned how to:

- Configure remote logging for a logging server
- Redirect system log messages to a centralized console server



UNIT THIRTEEN

WEB SERVICE

Introduction

Topics covered in this unit:

- Deploy SSL web services
- Configure a web server with virtual hosts
- Configure a web server with dynamic content
- Configure a web server with authenticated directories

Securing Apache with Encryption

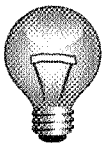
Apache HTTP Server Configuration Review

You should already know the basics of how to set up a simple **Apache HTTP Server**. In this unit, we will take a closer look at some more advanced but common configurations used with web server deployments, starting with how to set up support for TLS/SSL connections.

Apache HTTP Server is installed by the **web-server** yum group, the most critical package being *httpd*, which provides the core components of the web server. The **httpd** service script is used to start and start the server, and it listens for connections on TCP port 80 by default. It is confined by the SELinux policy for added security, which we have discussed and which is documented in the **httpd_selinux(8)** man page. Its main configuration file is **/etc/httpd/conf/httpd.conf**, which automatically includes all the files matching **/etc/httpd/conf.d/*.conf** as part of its file. By default, web content is served from subdirectories of **/var/www**, with the "**DocumentRoot**" of actual web pages in **/var/www/html**, although this can be changed in the configuration file.

Steps to Deploy TLS/SSL Encryption

Support for TLS/SSL web sites is provided by the *mod_ssl* RPM package. Its configuration file is **/etc/httpd/conf.d/ssl.conf**. Simply by installing it and restarting the web server, a SSL encrypted version of the default website on the server will be made available with a self-signed test certificate for localhost.



Important

To contact a TLS/SSL encrypted website on your server using a **https://** URL, you must make sure that clients can connect to TCP port 443 on your web server. Check your firewall settings.

If you use a web browser to connect to the secure web site, you will probably get a warning that the SSL certificate for the web site does not match the hostname of the site, and that the certificate is not signed by a trusted Certificate Authority. To fix this, you will need to obtain an SSL certificate for your web site's hostname signed by a public Certificate Authority, and you will need to make some configuration changes to **/etc/httpd/conf.d/ssl.conf**. The critical directives are **SSLCertificateFile**, which should point to a file in **/etc/pki/tls/certs/** containing your public SSL certificate, and **SSLCertificateKeyFile**, which should point to a file in **/etc/pki/tls/private/** containing your private SSL key. We will not go into further depth in this class on how to obtain a signed SSL certificate.

So, the basic steps are:

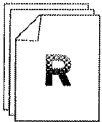
1. Make sure the **web-server** yum group is installed:

```
yum groupinstall web-server
```

2. Install the *mod_ssl* package:

```
yum install mod_ssl
```

3. If you are replacing the test certificate with a signed one:
 - Copy your certificate and private key to appropriate places in **/etc/pki/tls/**
 - Make sure both files have the SELinux type **cert_t** and that the private key is not world readable
 - Open **/etc/httpd/conf.d/ssl.conf** in an editor
 - Point **SSLCertificateFile** at your SSL certificate
 - Point **SSLCertificateKeyFile** at your SSL private key
 - Save and exit editing **/etc/httpd/conf.d/ssl.conf**
4. Restart the **httpd** service



References

Red Hat Enterprise Linux Deployment Guide

- Section 11.6: Setting Up an SSL Server

Apache.org: "Apache TLS/SSL Encryption"

<http://httpd.apache.org/docs/2.2/ssl/>

(if **httpd-manual** is installed and **httpd** is running):

<http://localhost/manual/ssl/>



Practice Performance Checklist

Apache mod_ssl Basics

Deploy an SSL encapsulated Apache web server on serverX. It should use the default self-signed SSL certificate.

- ☐ Log into serverX as root.
- ☐ Install the Apache web server (**httpd**) package, if necessary.
- ☐ Install the **mod_ssl** package.
- ☐ Examine the **/etc/httpd/conf.d/ssl.conf** configuration file provided by the **mod_ssl** package.
 - What is the Apache directive that points to the SSL certificate?
 - What is its value?
- ☐ Restart the **httpd** service.
- ☐ Launch Firefox and browse to **https://serverX.example.com**. When Firefox presents a warning, take further steps to examine the certificate with Firefox.
 - Click the "I Understand the Risks" link.
 - Click the "Add Exceptions..." button, then click "View..." when it becomes active.
 - Browse the information presented in both the "General" and "Details" tabs.
 - Click "Close" when you are finished inspecting the certificate information.

Configure Name-Based Virtual Hosting

Virtual hosts allow you to serve multiple web sites at the same time from a single **httpd** server. In this section we will look at *name-based virtual hosts*, in which multiple hostnames all point to the same IP address, but the web server delivers a different web site with different content depending on the hostname used to reach the site.

The way this will work is that a particular IP address or addresses on the web server will be identified as being shared by the name-based virtual hosts. The web server will look in its configuration files for the first **VirtualHost** block for the IP address of each incoming request in which the **ServerName** or **ServerAlias** matches the hostname used by the request. It will serve content based on the configuration of that matching **VirtualHost** block. If the hostname does not match in any block, then the first **VirtualHost** block for the IP address of the request is used by default.



Important

When you add virtual hosts to your web server configuration, you need to make sure you have a **VirtualHost** block that matches your original main website if you intend to keep serving it out. The directives for the main website are used as *defaults* for the virtual hosts, which may be overridden by each **VirtualHost** block.

The following is a list of key concepts and settings you will need to know in order to set up name-based virtual hosts. Pay attention as the class discusses each one and take notes of where, when, and how you would use each along with any examples that the instructor gives. You will be asked to set up virtual hosts after this discussion.

```
# Name-based virtual hosts defined in /etc/httpd/conf/httpd.conf
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName www.wonka-chocolates.com
    ServerAlias wonka-chocolates.com
    ServerAdmin webmaster@wonka-chocolates.com
    DocumentRoot /var/www/wonka-chocolates.com/html
</VirtualHost>
```

- VirtualHost
- NameVirtualHost
- ServerName/ServerAlias

- `ServerAdmin`
- `DocumentRoot`
- `semanage fcontext`

Configuring one **DocumentRoot** to be administered by a group:

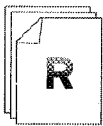
It is useful to use set-GID directories to make it easier for a group of web administrators to manage content under a **DocumentRoot**. Use `chgrp -R webadmins DocumentRoot` to set all the files in **DocumentRoot** to be owned by group *webadmins*, that is, whatever group your webadmins are all in. Then make sure that group has set-GID on **DocumentRoot**: `chmod 2775 DocumentRoot`. (A fancier way to set set-GID and write for the group on *all* subdirectories of **DocumentRoot** is `find DocumentRoot -type d -exec chmod g+ws '{}' \;`, where *DocumentRoot* is replaced with the actual **DocumentRoot** directory.)

You should also ensure that the SELinux type on the contents of **DocumentRoot** is either `httpd_content_t` or `public_content_t` to allow the web server to serve the contents out.



Note

There is another type of virtual host, the *IP-based virtual host*, in which each virtual host has its own IP address on the server. These types of virtual hosts work better with TLS/SSL sites; note that the default SSL service configured in `/etc/httpd/conf.d/ssl.conf` is an IP-based virtual host. For more information on IP-based virtual hosts, see the **Apache HTTP Server** documentation.



References

Red Hat Enterprise Linux Deployment Guide

- Section 11.5: Virtual Hosts

Red Hat Enterprise Linux Deployment Guide

- Section 15.5.1: Group Directories

Apache.org: "Name-Based Virtual Host Support"

<http://httpd.apache.org/docs/2.2/vhosts/name-based.html>

(if **httpd-manual** is installed and **httpd** is running)

<http://localhost/manual/vhosts/name-based.html>



Practice Performance Checklist

Configure Name-Based Virtual Hosts

For this exercise, `wwwX.example.com` is already set up as a **CNAME** alias to `serverX.example.com`.

When you finish the checklist, you will run a grading script, so make sure your web server serves up the content exactly as described in the steps.

- ☐ Create `/var/www/html/index.html` containing the text "**this is serverX.**"
- ☐ From `desktopX`, use Firefox to verify that the websites `wwwX`, `wwwX.example.com`, `serverX`, and `serverX.example.com` all display your custom **index.html**.
- ☐ Create `/wwwX/html/index.html` containing the text "**this is wwwX.**"
- ☐ Modify Apache to enable name-based virtual hosting. `serverX` and `serverX.example.com` should serve `/var/www/html/index.html` as the main page. `wwwX` and `wwwX.example.com` should serve `/wwwX/html/index.html` as the main page.
- ☐ Do not disable SELinux (Hint: You may need to modify the SELinux file context database, or change the SELinux type of certain files).
- ☐ When you finish, run the **lab-grade-virthost** evaluation script from `serverX` to make sure you have done everything correctly.

Stage a CGI executable

CGI, the Common Gateway Interface, is the easiest way to put dynamic content on a web site. The web server acts as an application gateway to a *CGI script* which runs on the server and generates HTML output in its response which the server sends back to the browser.

CGI scripts can be used for many purposes, but it is important to carefully control which CGI scripts are used and who is permitted to add and run them. A poorly written CGI script may provide a way for an external attacker to compromise the security of the web site and its contents. Therefore, both at the web server level and at the SELinux policy level there are settings which are used to restrict use of CGI scripts.

Install a local copy of the How-To by installing the **httpd-manual** RPM with **yum** on serverX. Restart Apache after you install the package to make the Apache documentation available for browsing. Consult the Apache.org "Dynamic Content with CGI" tutorial in the References below to see examples and sample commands that perform the following steps.

Given a CGI script to be located at **/wwwX/cgi-bin/hostinfo.cgi**, what configuration syntax, filesystem permissions, and SELinux context type would you need to know?

1. Create a directory outside of the web site's **DocumentRoot**:

```
[root@serverX ~]# mkdir -p /wwwX/cgi-bin
```

2. Configure Apache to recognize it as a source for CGI programs:
3. Change the SELinux context of the CGI script directory to **httpd_sys_script_exec_t**:
4. Restart Apache to put the changes into effect:

Make sure you understand how to enable CGI on a directory if someone were to give you a CGI script.

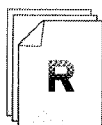
1. Copy the script into the CGI script directory:
2. Ensure the script is not writable by the **httpd** daemon:
3. Make it executable:



Warning

One of the most common sources of security issues that lead to web site compromises are software bugs in web applications. Be careful when writing code for CGI applications!

CGI programs must be written in a way they will produce content the Apache web server expects. Writing scripts that produce valid CGI output is the responsibility of web developers and is beyond the scope of this course.



References

Apache.org: "Apache Tutorial: Dynamic Content with CGI"

<http://httpd.apache.org/docs/2.2/howto/cgi.html>

(if **httpd-manual** is installed and **httpd** is running)

<http://localhost/manual/howto/cgi.html>

Red Hat Enterprise Linux Managing Confined Services

- Chapter 3: The Apache HTTP Server

httpd_selinux(8) man page



Practice Quiz

Apache CGI Quiz

1. CGI stands for ,
(select one of the following...)
 - a. Content Generated Interface
 - b. Command Gateway Interface
 - c. Common Generated Interface
 - d. Common Gateway Interface

2. The last argument in **ScriptAlias** `/cgi-bin/ /my/private/cgi-bin/` is
(select one of the following...)
 - a. relative to **DocumentRoot**
 - b. relative to **/var/www/**
 - c. relative to **ServerRoot**
 - d. an absolute path on the filesystem

3. The default **ScriptAlias** in `/etc/httpd/conf/httpd.conf` is pointing to ...
(select one of the following...)
 - a. `/var/www/cgi-bin`
 - b. `/var/html/cgi-bin`
 - c. `/cgi-bin`
 - d. `/var/www/html/cgi-bin`

4. One of the built in SELinux context types for a generic CGI programs is
(select one of the following...)
 - a. **httpd_t**
 - b. **httpd_sys_script_exec_t**
 - c. **script_t**
 - d. **httpd_content_t**

5. The Apache process should have the following filesystem permissions on CGI programs
(select one of the following...)
 - a. **---**
 - b. **r--**
 - c. **r-x**
 - d. **rwX**

Configure User-Based Authentication

It can be useful to limit access to parts of a web site to authorized users only. One way to do this is to configure *user-based authentication* based on usernames and passwords, where the web server prompts the user to login when certain pages are accessed.

There are a number of different ways to implement this. We will look at two in this course: *flat-file authentication* where users are defined in a local password file, and *LDAP authentication*, where users are defined in a remote LDAP directory server.

Your instructor will demonstrate how to set up user-based authentication. Take notes as you will be asked to configure your web server to do the same in lab.

Apache Flat-file User Authentication

In this configuration, user accounts and passwords are stored in a local **.htpasswd** file. For security reasons, this file should not be kept in the **DocumentRoot** of the web site, but should be in some directory the web server does not serve out. The special **htpasswd** command is used to manage users in the **.htpasswd** file.

Example configuration procedure:

- Create an Apache password file with two accounts:

```
[root@serverX]# htpasswd -cm /etc/httpd/.htpasswd bob
[root@serverX]# htpasswd -m /etc/httpd/.htpasswd alice
```

- Assuming that the **VirtualHost** block has been defined previously, add a section such as the following into the **VirtualHost** block:

```
<Directory /var/www/virtual/wwwX/html>
    AuthName "Secret Stuff"
    AuthType basic
    AuthUserFile /etc/httpd/.htpasswd
    Require valid-user
</Directory>
```

- Test access using a web browser; verify that access works for authenticated users but fails otherwise

Apache LDAP User Authentication

In this configuration, user accounts and passwords are stored in a remote LDAP directory service. The advantage of this configuration is that multiple web servers can use the same directory service to store user accounts and passwords, making it easier to keep them synchronized. In order to set this up, you need to know the location of the LDAP server that contains your account information, whether it uses TLS/SSL, and what LDAP prefix your user entries are under. Your LDAP administrator will need to appropriately manage account information in the directory. For the purposes of this course, we will focus on the web server part of this configuration and ignore how information is managed in the LDAP directory.

Example configuration procedure:

- Download the LDAP certificate, if necessary. In class it can be downloaded from `ftp://instructor.example.com/pub/example-ca.crt`
- Add **LDAPTrustedGlobalCert CA_BASE64** `/etc/httpd/example-ca.crt` and **AuthBasicProvider ldap** to `httpd.conf`.
- Add a **Directory** block inside the **VirtualHost** block just like you did for flat-file authentication above. The **AuthUserFile** line, however, should be replaced with an **AuthLDAPUrl** line pointing to the LDAP directory to search:

```
AuthLDAPUrl "ldap://instructor.example.com/dc=example,dc=com" TLS
```

for example,

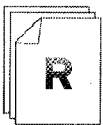
```
LDAPTrustedGlobalCert CA_BASE64 cert-path

<Directory /var/www/html/private>
    AuthName "A very private place"
    AuthType basic
    AuthBasicProvider ldap
    AuthLDAPUrl "ldap://fqdn/prefix" TLS
    Require valid-user
</Directory>
```

where

- *cert-path* is the pathname of CA certificate
 - *fqdn* is the fully qualified domain name of the LDAP server
 - *prefix* is the LDAP prefix (**dc=example, dc=com** for example)
- Test that LDAP users can authenticate to Apache.

Use this space for notes



References

Apache.org: "Authentication, Authorization and Access Control"

<http://httpd.apache.org/docs/2.2/howto/auth.html>

(if the **httpd-manual** and **httpd** is running)

<http://localhost/manual/howto/auth.html>

Apache Module `mod_authnz_ldap` documentation

http://localhost/manual/mod/mod_authnz_ldap.html



Practice Performance Checklist

Configure LDAP-Based Authentication

You will configure the web server on serverX with a **/private** URL that is accessible by users in the LDAP directory on instructor.example.com.

- ☐ Configure LDAP authentication on serverX using instructor.example.com as the LDAP server, **dc=example,dc=com** for the base distinguished name and use the certificate found at *ftp://instructor/pub/example-ca.crt*. Choose LDAP passwords.
- ☐ Login as **root** on serverX. Create a new directory **/var/www/html/private**.
- ☐ In the **private** directory, create an **index.html** containing the text **Private Data**
- ☐ Download **ftp://instructor/pub/example-ca.crt** and place it in **/etc/httpd**
- ☐ Edit **/etc/httpd/conf/httpd.conf** and add LDAP authentication for the **private** directory.

```
LDAPTrustedGlobalCert CA_BASE64 /etc/httpd/example-ca.crt

<Directory /var/www/html/private>
    AuthName "Secret Stuff"
    AuthType basic
    AuthBasicProvider ldap
    AuthLDAPUrl "ldap://instructor.example.com/dc=example,dc=com" TLS
    Require valid-user
</Directory>
```

- ☐ Restart Apache
- ☐ Browse to **http://serverX.example.com/private**. You should see an authentication dialog box pop up. If not, close all browser windows, check your configuration, and try again.
- ☐ Log in as user **ldapuserX** with a password of **password**

Troubleshooting Apache SELinux issues

- List current port SELinux contexts

```
[root@serverX ~]# semanage port -l | grep http
http_cache_port_t      tcp      3128, 8080, 8118, 10001-10010
http_cache_port_t      udp      3130
http_port_t            tcp      80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t    tcp      5988
pegasus_https_port_t   tcp      5989
```

- Assign a port an SELinux context

If you change Apache to run on a non-standard port, you will likely need to assign that port the **http_port_t** SELinux context. For example, if you changed Apache to run on port 777, you would do the following to allow Apache to run:

```
[root@serverX ~]# semanage port -a -t http_port_t -p tcp 777
```

- Apache log files and logging levels

You can define the log level and define the log files using **LogLevel**, **ErrorLog** and **CustomLog** in **/etc/httpd/conf/httpd.conf**. The default **LogLevel** is **warn**. By default, the **ErrorLog** is sent to **/var/log/httpd/error_log** and **CustomLog** is sent to **/var/log/httpd/access_log**. These directives can be used to define directives for the main web site, or for virtual hosts.

- Make SELinux more verbose, SELinux log files

There are rules in the SELinux policy that prevent error messages from being sent to the logs. These rules are called **dontaudit** rules. They prevent the logs from filling with useless messages, but they can prevent you from determining an issue if you are troubleshooting. To disable these **dontaudit** rules, run the following command:

```
[root@serverX ~]# semanage dontaudit off
```



Warning

Disabling the **dontaudit** rules will also disable **setroubleshoot-server**. No messages will be sent to **/var/log/messages**, and **sealert** will be disabled. All of the SELinux error messages will be sent to **/var/log/audit/audit.log**.

- Restore or change the SELinux context type of an html file or directory

If you are storing web data in a new location, use **semanage fcontext** to add new location to the context database and use **restorecon** to set the contexts of the files/directories:

```
[root@serverX ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
```



```
[root@serverX ~]# restorecon -RFvv /virtual/
```

If the content is in a known-good location (e.g., `/var/www/html/`), but you are getting permission denied errors, run the **restorecon** command on the directory as above. The **restorecon** command can take the **-F** option which will change the *customizable* types that **restorecon** normally ignores (`/etc/selinux/targeted/contexts/customizable_types` contains this list of types).



Note

If you get permission denied errors, remember that it could be a Linux permission issue, not an SELinux permission issue. Make sure that the **apache** user or group has at least read access to the files and directories in question.

- SELinux documentation on Apache context types and booleans

The **httpd_selinux(8)** man page includes descriptions of the most common file contexts and booleans for the web server.

httpd_sys_content_t is used for any general file/directory for the web server.
httpd_sys_script_exec_t is used for scripts (e.g., CGI) executed by the web server.
public_content_t is the context for files that will be shared with other services that are being restricted by SELinux such as FTP, rsync, Samba, etc.

- List Apache SELinux booleans

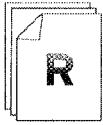
Use the **getsebool** command to view booleans:

```
[root@serverX ~]# getsebool -a
abrt_anon_write --> off
allow_console_login --> on
allow_corosync_rw_tmpfs --> off
allow_cvs_read_shadow --> off
...
[root@serverX ~]# getsebool httpd_enable_cgi
httpd_enable_cgi --> on
```

- Make SELinux booleans persist

To make booleans persistent, use **setsebool -P**:

```
[root@serverX ~]# setsebool -P httpd_enable_cgi off
[root@serverX ~]# getsebool httpd_enable_cgi
httpd_enable_cgi --> off
[root@serverX ~]# semanage boolean -l | grep httpd_enable_cgi
httpd_enable_cgi          -> off      Allow httpd cgi support
```



References

Red Hat Enterprise Linux SELinux Guide

- Section 5.6: Booleans

Red Hat Enterprise Linux SELinux Guide

- Chapter 8: Troubleshooting

Red Hat Enterprise Linux Managing Confined Services

- Chapter 3: The Apache HTTP Server

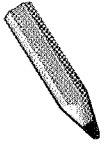
semanage(8), **httpd_selinux(8)** man pages



Practice Quiz

Troubleshooting Apache Quiz

1. Complete the following command to list all port contexts:
semanage _____ **-l**.
2. Complete the following command to enable Apache to use TCP port 8001: **semanage** _____ **-t**
httpd_port_t _____ **8001**.
3. The two Apache config file directives to specify the severity (how verbose) error messages are and which file to write to are _____ and _____.
4. The Apache config file directive to specify the format and location of clients accessing content is _____.
5. Full (raw) SELinux AVC messages go to **/var/**
log _____.
6. To make SELinux more verbose, you can run **semanage**
_____ **off**.
7. The commands to get and set SELinux booleans are _____ and _____.
8. **man** _____ will present an SELinux man page specific to Apache.
9. **man -k** _____ lists all service specific SELinux man pages.
10. The **-F** option to **restorecon** will reset _____ types.



Test

Criterion Test 1

Case Study

SSL Encapsulated Web Services

Before you begin...

Run the **lab-setup-hacker** script on desktopX.

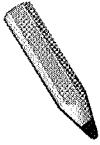
Marcelo Hacker is a successful private investigator. In fact, he is doing so well that it is becoming difficult to find the time to meet with prospective clients. Mr. Hacker has decided to set up a website where prospective clients can send him messages. As confidentiality is an important part of the private investigation business, the website must use a signed SSL certificate.

- Set up Apache on serverX to provide an SSL encrypted website for Marcelo Hacker.
- A signed SSL server certificate for your server and a matching key can be found at the following location: **/net/instructor/var/ftp/pub/materials/tls**. Below that directory **certs/serverX.crt** contains the signed certificate for your server and **private/serverX.key** has the private key that matches it.

Deploy the signed certificate for Apache on serverX. Leave the default placeholder website for content. Mr. Hacker will upload his custom content at a later date.

When you have completed the requirements, run **lab-grade-hacker** script on desktopX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Test

Criterion Test 2

Case Study

Web Server Additional Configuration

Before you begin...

Run the **lab-setup-website** script on desktopX.

Example Industries, a fine example of a company, needs a new website. In fact, they need two! One will be the company website and the other will be for testing content. Additionally, the company website will need a password protected area and a special CGI application installed.

On your serverX machine, deploy a web server with two virtual hosts.

Virtual host 1: `http://serverX.example.com`

- Create a simple placeholder page for the base URL

Virtual host 2: `http://wwwX.example.com`

- Create a simple placeholder page for the base URL that is different from the one used on virtual host 1
- Make `http://wwwX.example.com/private` a password protected area
- Add user **forrest** with password **trees** to **/private**
- Download the CGI file `ftp://instructor.example.com/pub/gls/special.cgi` and install it as `http://wwwX.example.com/cgi-bin/special.cgi`

When you are ready to check your work, run the grading script **lab-grade-website** on desktopX.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Securing Apache with Encryption

In this section you learned how to:

- Install a certificate for the Apache web server and enable SSL encapsulation

Configure Name-Based Virtual Hosting

In this section you learned how to:

- Configure a name-based virtual host (separate DocumentRoot, ServerAdmin, ServerName)
- Establish a ServerAlias
- Configure one DocumentRoot to be managed by a group

Stage a CGI executable

In this section you learned how to:

- Stage a CGI executable

Configure User-Based Authentication

In this section you learned how to:

- Limit information to a small group of users using flat-file authentication
- Limit information to a centralized group of users using an LDAP server or relational database

Troubleshooting Apache SELinux issues

In this section you learned how to:

- Monitor and audit web activity
- Add a persistent SELinux file context mapping to the policy
- Modify SELinux to allow a service to use a non-standard port
- Troubleshoot SELinux context and boolean conflicts

Criterion Test 2

In this section you learned how to:

-



UNIT FOURTEEN

BASIC SMTP CONFIGURATION

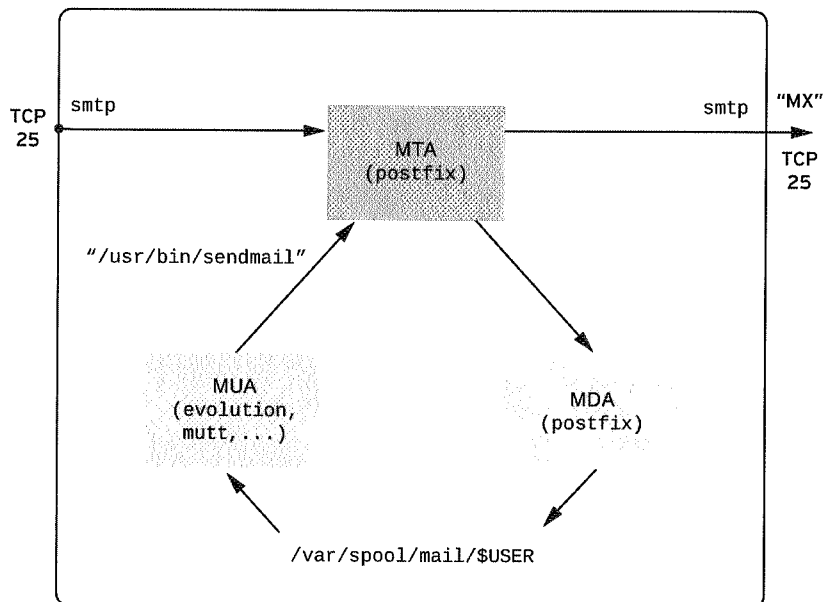
Introduction

Topics covered in this unit:

- Basic e-mail configuration
- Intranet server configuration

Basic E-mail Delivery

E-mail Delivery



- **MTA:** "Mail Transfer Agent". MTAs relay mail from point to point until it can be delivered. E-mail is submitted by other servers using the SMTP protocol to TCP port 25, or by local clients with the `/usr/bin/sendmail` program. If the MTA is the final destination, the message is passed to the MDA. If not, it looks up the next MTA in DNS using MX records and relays it there using SMTP.
- **MDA:** "Mail Delivery Agent". The MDA delivers mail to the recipient's local message store (by default, `/var/spool/mail/user`). Postfix provides its own MDA to deliver to the default local file-based message store, `/usr/libexec/postfix/local`.
- **MUA:** "Mail User Agent". Clients used to send e-mail and read e-mail in the user's message store.

Key e-mail delivery concepts:

- **Relaying:** when an e-mail server (MTA) forwards submitted mail to another server for delivery
- **Queueing:** a failed delivery or relay attempt is queued and retried periodically by the MTA. (By default Postfix does this once every hour.)
- **Rejected:** when an e-mail message is refused by an e-mail server during the initial submission
- **Bounced:** when an e-mail message is returned by a remote server to the originating e-mail server and/or user after it has been accepted for delivery by the remote server

The Postfix MTA

A number of open source e-mail servers exist, including Postfix, Sendmail, and Exim. In this unit, we will focus on Postfix, a powerful but relatively easy to configure MTA, which happens to be used by default in Red Hat Enterprise Linux 6.



Note

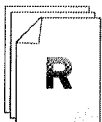
Sendmail was the default MTA in Red Hat Enterprise Linux 5 and earlier.

Postfix is provided by the *postfix* RPM package, and is controlled by the **postfix** service script. It is a modular program made up of several cooperating programs, its components controlled by the **master** process.

The main configuration file for Postfix is **/etc/postfix/main.cf**, which may be edited using a text editor or with the **postconf** command. The **postconf** command may also be used to determine current and default configuration settings, either for all of Postfix or on an option-by-option basis.

By default, Postfix only listens for incoming e-mail from localhost. To reconfigure Postfix to receive mail for local delivery sent from remote hosts, **inet_interfaces = all** must be set in **/etc/postfix/main.cf**.

When troubleshooting e-mail, a log of all mail-related operations is kept in **/var/log/maillog**, which includes information about rejections and successful deliveries. The **mailq** command (or **postqueue -p**) displays a list of any outgoing mail messages that have been queued. To attempt to deliver all queued messages again immediately, you can run the **postfix flush** command (or **postqueue -f**); otherwise Postfix will attempt to resend them about once an hour until they are accepted or expire.



References

Red Hat Enterprise Linux Deployment Guide

- Section 12.3.1: Postfix

postconf(5) and **postfix(1)** man pages



Practice Performance Checklist

Basic E-mail Delivery

- ☐ Start two terminal shells, one as **root** on desktopX, the other as **root** on serverX.
- ☐ On each machine, confirm that **postfix** is installed and that the **postfix** service is running.
- ☐ On serverX, add the user **elvis**. Become **elvis** and open the **mutt** MUA to monitor Elvis' incoming mail.
- ☐ Use the **mutt** MUA on desktopX to compose and send mail to **elvis@serverX.example.com**.
- ☐ Did **elvis** receive e-mail on serverX? Hmm...
- ☐ On desktopX use **mailq** to examine the delivery queue. Also browse **/var/log/maillog** to look for any problems.
- ☐ On serverX, recall that **postfix** binds to **localhost** only by default according to Red Hat policy. Use **netstat** to confirm this is the case.
- ☐ Examine the current settings of the **inet_interfaces** directive in the main **postfix** configuration file **/etc/postfix/main.cf**.
- ☐ Edit **/etc/postfix/main.cf** and set **inet_interfaces=all**. Restart the **postfix** service and confirm the daemon is listening on all interfaces..
- ☐ On desktopX use **postfix flush** to manually flush the pending delivery queue. Confirm that the queue is now empty and that the mail was successfully delivered.
- ☐ On serverX use the **mutt** MUA to confirm e-mail delivery. Also examine **/var/log/maillog** for evidence the mail was successfully delivered.

Intranet Configuration

In practice, most organizations do not have one mail server that handles all inbound and outbound e-mail any more. Instead, mail servers are specialized for particular roles for security reasons and so that they may better tune performance for their particular intended applications.

Some standard roles include:

- *null client*: A client machine that runs a local MTA, but only so that all e-mail can be forwarded to a central mail server for delivery. A null client does not accept local delivery for *any* e-mail messages. Users may run MUAs on the null client to read and send e-mail messages. Most machines will be null clients.

In the diagram below, we use desktopX.example.com to stand in for all null clients.

- *inbound-only mail server*: A mail server that handles all incoming e-mail for users at the site and hands it to an MDA for delivery to user message stores. Outbound mail is forwarded to a central mail server in the same way as a null client. The inbound mail server may be integrated with an IMAP or POP3 server to allow user MUAs to access their message stores, or a separate host with access to the message store may run the IMAP or POP3 server.

In a real-world scenario, there is usually an anti-spam mail server or appliance in front of the inbound-only mail server that filters out spam and only relays good messages to the inbound mail server. We will not discuss configuration of this device in this course.

In the diagram below, mail.example.com is the inbound-only mail server, and it also runs the IMAP server.

- *outbound mail relay*: The outbound mail relay, or "smarthost", accepts all outbound messages, and using MX records and the SMTP protocol relays them toward their destination. An outbound mail relay must only relay e-mail for authorized hosts; an "open mail relay" will be exploited by spammers, and other mail servers will likely block messages from a known open relay.

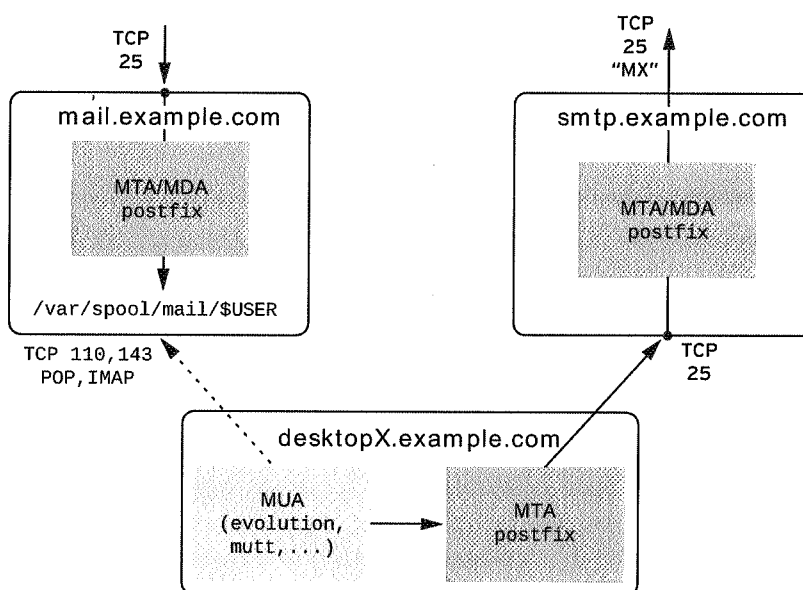
For simplicity, in this class we will look at an internal workgroup outbound relay, which will accept messages sent to port 25/TCP from internal IP addresses for any destination without further authentication, but which does not accept messages from external IP addresses.

In the diagram below, smtp.example.com is the outbound mail relay used by the null clients.



Note

A more sophisticated variation on the outbound mail relay is the *mail submission agent*. An MSA relays messages from internal or external machines which have been authenticated by username and password over an SSL-protected SMTP connection to port 587/TCP. This allows users not on the local network to relay e-mail securely through the outbound mail relay. It is more complex to configure, and we will not discuss it further in this class.



External DNS

example.com. IN MX 10 mail.example.com.

Concept	Directive	mail.example.com	desktop.example.com	smtp.example.com
Binding Interface	<i>inet_interfaces</i>			
Masquerade as	<i>myorigin</i>			
Indirect Delivery	<i>relayhost</i>			
Receive mail for ...	<i>mydestination</i>			
Local Delivery	<i>local_transport</i>			
Relay from ...	<i>mynetworks</i>			

Table 14.1. Intranet Mail Server, Null Client, and Outbound Relay

Important Postfix configuration directives

These are all found in the `/etc/postfix/main.cf` file.

inet_interfaces

Controls which network interfaces Postfix listens on for incoming e-mail. If set to **loopback-only** listens only on 127.0.0.1 and ::1, and if set to **all** listens on all network interfaces. Can also specify particular addresses.

myorigin

Rewrite locally posted e-mail to appear to come from this domain. (Helps ensure responses return to the inbound mail server.)

relayhost

The smarthost to relay all outbound mail through. Normally given in square brackets to suppress MX record lookup.

mydestination

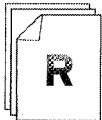
E-mail addressed to these domains are passed to the MDA for local delivery.

local_transport

How messages addressed to **\$mydestination** should be delivered. By default, set to **local:\$myhostname**, which uses the **local** MDA to deliver incoming e-mail to the local message store in `/var/spool/mail`.

mynetworks

Comma-separated list of IP addresses and networks (in CIDR notation) that can relay through this MTA to anywhere, without further authentication.



References

postconf(5), **postconf(1)**, and **transport(5)** man pages

`/usr/share/doc/postfix-*/README_FILES/BASIC_CONFIGURATION_README`

`/usr/share/doc/postfix-*/README_FILES/STANDARD_CONFIGURATION_README`



Practice Case Study

Intranet Configuration

Before you begin...

DNS has already been configured to overlay your hosts as members of the **domainX.example.com** domain.

hostname	ip address	also known as
mail.domainX.example.com	192.168.0.X+100	(serverX.example.com)
smtp.domainX.example.com	192.168.0.X+200	(hostX.example.com)
desktop.domainX.example.com	192.168.0.X	(desktopX.example.com)

Table14.2. domainX.example.com

Also, the host mail.domainX.example.com is the **MX** recipient for the entire domainX.example.com domain.

Complete the following table with the appropriate directives to configure these hosts to act as an intranet mailbox server, smtp host, and client station, respectively.

Try to use only the the **BASIC_CONFIGURATION_README**, **STANDARD_CONFIGURATION_README** and **main.cf** files for reference.

Once complete, have another student review your work.

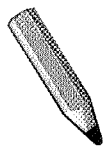
External DNS

domainX.example.com. IN MX 10 mail.domainX.example.com.

Concept	Directive	mail.domainX	desktop.domainX	smtp.domainX
Binding Interface	<i>inet_interfaces</i>			
Masquerade as	<i>myorigin</i>			
Indirect Delivery	<i>relayhost</i>			
Receive mail for ...	<i>mydestination</i>			
Local Delivery	<i>local_transport</i>			
Relay from ...	<i>mynetworks</i>			

Table14.3. Intranet Mail Configuration for domainX.example.com

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Test

Criterion Test

Case Study

Intranet E-mail Configuration

Before you begin...

Before you begin, run the script **lab-setup-email** on desktopX

The Hoffman Hair Supply company, a manufacturer of hair grooming products, wants to centralize the management of their internal e-mail.

DNS has already been configured so that your machines are members of the DNS domain **domainX.example.com** with the following addresses:

192.168.0.X	desktop.domainX.example.com	(a.k.a. desktopX.example.com)
192.168.0.X+100	mail.domainX.example.com	(a.k.a. serverX.example.com)
192.168.0.X+200	smtp.domainX.example.com	(a.k.a. hostX.example.com)

Also, the **mail.domainX.example.com** server is the **MX** recipient for the entire **domainX.example.com** domain.

Configure the **mail.domainX.example.com** host to act as an incoming mail-only server, so that all mail delivered to the **@domainX.example.com** domain is stored on this server.

Configure the **smtp.domainX.example.com** server to act as an outgoing SMTP server, which is willing to relay mail from members of the **domainX.example.com** domain to outside networks.

Configure the **desktop.domainX.example.com** host to act as a "null client". It cannot receive e-mail from the network, local mail delivery is disabled, and all outgoing e-mail is sent indirectly via **smtp.domainX.example.com**.

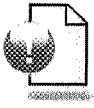
For all three hosts, make sure that any originating mail masquerades the sender's domain as **domainX.example.com**.

When you are finished run the **lab-grade-email** script to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Basic E-mail Delivery

In this section you learned how to:

- Configure postfix to receive mail from the network for system administrative tasks.
- Monitor e-mail delivery, and diagnose potential delivery problems.

Intranet Configuration

In this section you learned how to:

- Determine if a mailbox server and/or an SMTP server is appropriate.
- Masquerade the domain of sender's addresses.
- Relay all mail to domain mail server.
- Disable local delivery.
- Relay mail only for local network.



UNIT FIFTEEN

CACHING-ONLY DNS SERVER

Introduction

Topics covered in this unit:

- Non-authoritative DNS servers

DNS Overview

In this section the class will review what you already know about DNS and ensure that you understand the types of DNS servers, DNS resource records, and the basics of DNS operation. Ask questions and take notes here as there will be a game after the class discussion.

Authoritative nameservers

Stores and serves actual data for a zone (all or part of a DNS domain). Types of authoritative nameservers include:

- *Master*: contains original zone data. Sometimes called a "primary" nameserver.
- *Slave*: backup server which gets copies of zone data from the master through *zone transfers*. Sometimes called a "secondary" nameserver.

Non-authoritative / recursive nameservers

Used by clients to look up data from authoritative name servers. Types of recursive nameservers include:

- *Caching-only nameserver*: only used for lookups, not authoritative for anything but trivial data

DNS Lookups

- *Stub resolver* on client sends query to the nameserver in `/etc/resolv.conf`
- If the nameserver is *authoritative* for the information requested, sends *authoritative answer* to client
- Otherwise, if the nameserver has the information requested in its cache, it sends a *non-authoritative* answer to the client
- If information is not in the cache, the nameserver searches for the authoritative nameserver for the information, starting with the root zone and working down the DNS hierarchy to the nameserver which is authoritative for the information, and gets the answer for the client. In this case the nameserver passes the information to the client and also keeps a copy in its own cache for future lookups.

Use this space for notes

DNS Resource Records

A DNS zone stores its information in the form of *resource records*. Each resource record has a *type*, which indicates what kind of data it holds:

- **A:** name to IPv4 address
- **AAAA:** name to IPv6 address
- **CNAME:** name to "canonical name" (another name with an A/AAAA record)
- **PTR:** IPv4/IPv6 address to name
- **MX:** mail exchanger for a name (where to send its e-mail)
- **NS:** name server for a domain name
- **SOA:** "start of authority", information for a DNS zone (administrative information)

Troubleshooting DNS lookups

The **dig** utility is one of the most useful utilities to troubleshoot issues with DNS lookups.

```
[student@student ~]$ dig example.com
; <<>> DiG 9.7.0-P2-RedHat-9.7.0-5.P2.el6 <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41645
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
server3.example.com.      IN      A

;; ANSWER SECTION:
server3.example.com.     86400   IN      A      192.168.0.103

;; AUTHORITY SECTION:
example.com.             86400   IN      NS      instructor.example.com.

;; ADDITIONAL SECTION:
instructor.example.com.  86400   IN      A      192.168.0.254

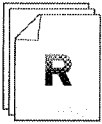
;; Query time: 2 msec
;; SERVER: 192.168.0.254#53(192.168.0.254)
;; WHEN: Mon Dec 13 10:06:48 2010
;; MSG SIZE rcvd: 94
```

It shows detailed information from a DNS lookup, including why a query may have failed:

- **NOERROR:** query was successful
- **NXDOMAIN:** DNS server says no such name exists
- **SERVFAIL:** DNS server is down or DNSSEC validation of response failed
- **REFUSED:** DNS server refuses to answer (perhaps for access control reasons)

Parts of **dig** output:

- The header indicates information about the query and answer, including the response status and any special flags that are set (**aa** for authoritative answer, and so on)
- **QUESTION:** the actual DNS query made
- **ANSWER:** the response, if any
- **AUTHORITY:** the nameservers responsible for the domain/zone
- **ADDITIONAL:** additional information provided, usually about the nameservers
- The comments at the bottom indicate which recursive nameserver was sent the query and how long it took to get a response



References

Red Hat Enterprise Linux Deployment Guide

- Section 10.1: Introduction to DNS

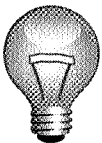
BIND Administrator's Reference Manual, Chapter 1
</usr/share/doc/bind-9.7.0/arm/>

host(1), **dig**(1), and **resolv.conf**(5) man pages

Caching-only DNS Servers

In this section, the class will learn how to configure BIND 9.7 as shipped in Red Hat Enterprise Linux 6 as a caching-only nameserver. Ask questions and take notes here on how to configure BIND as we will complete the game after discussion.

BIND is the most widely-used open source nameserver. In Red Hat Enterprise Linux it is provided by the *bind* software package. However, the main service it provides is the **named** program, which is controlled by the **named** service script.



Important

For BIND to work correctly, the firewall must allow connections to ports 53/UDP and 53/TCP on the nameserver. If 53/TCP is blocked, then large queries and zone transfers will probably break. This is a very common misconfiguration.

The main configuration file for BIND is **/etc/named.conf**. The **/var/named** directory contains additional data files used by the nameserver.

Syntax of **/etc/named.conf**:

- **//** or **#** to the end of a line is a comment; text between **/*** and ***/** is also a comment which can span multiple lines
- Directives end with a semi-colon **;**
- Many directives take *address match lists* as in curly braces; a list of IP addresses or subnets in CIDR notation, or named ACLs such as **any**; (all hosts) and **none**; (no hosts)
- The file starts off with an **options** block that contains directives that control how **named** works
- **zone** blocks control how **named** finds the root nameserver and the zones for which it is authoritative

Some important **options** directives:

- **listen-on** controls which IPv4 addresses **named** listens on
- **listen-on-v6** controls which IPv6 addresses **named** listens on
- **allow-query** controls which clients which clients can ask the DNS server for information
- **forwarders** contains a list of nameservers to which DNS queries will be forwarded (instead of directly contacting external nameservers; useful in firewalled scenarios)

All of these directives take semi-colon separated elements in curly braces as an address match list: **listen-on { any; };** or **allow-query { 127.0.0.1; 10.0.0.0/8; };** for example.

Configure a Caching-only Name Server Demonstration

- Install the *bind* software package:

```
[root@serverX]# yum install bind
```

- Edit **/etc/named.conf**:

```
listen-on port 53 { any; };  
listen-on-v6 port 53 { any; };  
allow-query { 192.168.0.0/24; };  
forwarders { 192.168.0.254; };
```

- Start and enable the DNS server:

```
[root@serverX]# service named start  
[root@serverX]# chkconfig named on
```



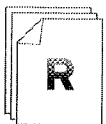
Note

The default configuration of BIND in Red Hat Enterprise Linux 6 integrates the settings formerly provided by the *caching-nameserver* package in Red Hat Enterprise Linux 5.



Important

BIND 9.7 in Red Hat Enterprise Linux 6 automatically attempts to validate DNS responses using DNSSEC by default. Since the root zone is signed, in a classroom without Internet access we have found that BIND may refuse to accept DNS responses from the classroom server because the real root nameservers cannot be contacted, causing DNSSEC validation errors. A workaround for this is to disable automatic DNSSEC validation by setting the option **dnssec-validation** to **no**.

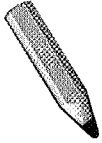


References

Red Hat Enterprise Linux Deployment Guide

- Unit 10: The BIND DNS Server

BIND Administrator's Reference Manual
`/usr/share/doc/bind-9.7.0/arm/`



Test

Criterion Test

Case Study

Caching-Only DNS Server

Before you begin...

Before you begin, run the script **lab-setup-cachingdns** on desktopX.

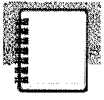
For his growing import/export business, Mr. Hnath would like to improve name resolution performance by deploying a caching name server at each of his business locations.

Recursive queries should be forwarded to the main name server at Hnath Import/Export headquarters.

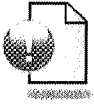
- Set up a caching name server on serverX.
- Configure the name server so that recursive queries are sent to instructor.example.com. Also, configure the name server to accept queries from anyone on the classroom network.

When you are ready, run **lab-grade-cachingdns** on desktopX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

DNS Overview

In this section you learned how to:

- Use **dig** to verify the functionality of your DNS server

Caching-only DNS Servers

In this section you learned how to:

- Install a caching nameserver
- Forward DNS requests to `instructor.example.com`



UNIT SIXTEEN

FILE SHARING WITH NFS

Introduction

Topics covered in this unit:

- NFS server configuration
- NFS client considerations

NFS Concepts and Configuration

NFS, the Network File System, is a network file system commonly used by Unix systems and network-attached storage filers to allow multiple clients to share access to files over the network. It may be used to provide access to shared directories of binaries, or to allow users to access their files from different clients in the same work group.

The NFS protocol comes in a number of versions: Linux supports versions 4, 3, and 2, with most system administrators being familiar with NFSv3. The protocol is insecure by default, but newer versions such as NFSv4 provide support for more secure authentication and even encryption with Kerberos. Fill in the table below as the instructor discusses some of the enhancements in NFSv4:

NFSv2	NFSv3	NFSv4
Original public NFS protocol	Extended NFSv2 architecture	
Still in use	Added features: TCP support 64-bit file sizes and offsets Larger read/write sizes	
	Some implementations (including Red Hat Enterprise Linux) support Kerberos	
Requires support services: <code>nfsd</code> , <code>rpc.mountd</code> , <code>rpc.statd</code> , <code>lockd</code>	Also requires support services: <code>nfsd</code> , <code>rpc.mountd</code> , <code>rpc.statd</code> , <code>lockd</code>	
More difficult to secure behind a firewall	More difficult to secure behind a firewall	
Useful for backward compatibility	Useful for backward compatibility	

NFS Server Configuration

To configure a basic NFS server, you should have the **nfs-file-server** package group (which includes the *nfs-utils* package) installed. You then should edit `/etc/exports` to list the file systems you intend to share to client systems over the network, and indicate which clients have what access to the export. For example:

```
/var/ftp/pub 192.168.0.0/24(ro, sync)
```

exports the directory `/var/ftp/pub` to all hosts on the 192.168.0.0/24 network with read-only permission. Likewise

```
/export/homes *.example.com(rw, sync)
```

exports the directory `/export/homes` with read-write permission to all hosts in example.com. Each export is specified on its own line in the server's `/etc/exports` file.

Any time you make an edit to **/etc/exports** when the NFS server is running, you should ensure these changes get applied by executing **exportfs -r** after saving your changes. You can use **exportfs -v** to display all exports.

NFSv4 also exports something called the *pseudo-root*, the root of all exported file systems. If a client mounts *nfs-server:/* this automatically mounts all exported file systems relative to their position under */* on the NFS server. This is useful for browsing all the file systems exported from a server on a client. You can still mount file systems individually as well.



Note

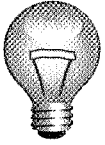
The **fsid=0** export option used in Red Hat Enterprise Linux 5 to manually export an NFSv4 pseudo-root is no longer necessary. The new NFS server automatically exports a pseudo-root without additional configuration or the need to configure bind mounts. (If for some reason you do specify the option, only the export with it set will be used as the top level of the pseudo-root.)

For file ownership and permissions to work properly, the user and group names that use NFS need to exist and be mapped consistently to the same UID and GID numbers on the clients and on the NFS server. These users can be manually configured in **/etc/passwd** and **/etc/group** using local tools, or you can keep them coordinate through a central LDAP authentication and user information directory.

By default, root on an NFS client is treated as user *nfsnobody* by the NFS server. That is, if root attempts to access a file on a mounted export, the server will treat it as an access by user *nfsnobody* instead. This is a security measure that can be problematic in scenarios where the NFS export is being used as */* by a diskless client and root needs to be treated as root. To disable this protection, the server needs to add **no_root_squash** to the list of options set for the export in **/etc/exports**:

```
/exports/root-192.168.0.1 192.168.0.1(rw,no_root_squash)
```

Note that this particular configuration is not secure, and would be better done in conjunction with Kerberos authentication and integrity checking (which is beyond the scope of this course).



Important

If user names have different UID numbers on different clients or the server, you will have ownership and permission problems.

One symptom of getting this wrong is if a file appears to be owned and usable by a particular non-privileged user, yet when they try to read or write it permission errors occur. The user probably has different UID numbers on the client and the server; using **ls -ln** on the file to show the owner's UID number and **id** to find the user's local UID number can help diagnose this issue. Another symptom of username/UID inconsistencies seen when using NFSv3 or older is that a user creates a file on a mounted NFS file system, but it shows up as being owned by a different user. Note that root is purposely mapped to a non-privileged user (nfsnobody) by default.

For NFSv4, port 2049/TCP (for **nfsd**) must be open on the server. For NFSv3 and earlier, additional ports must be opened for **rpcbind**, **rpc.mountd**, **lockd**, and **rpc.rquotad**, which is complicated by the fact that many of these services start on "randomly" selected ports. In addition, NFSv2 and NFSv3 support UDP transport, which requires appropriate ports to be opened as well. For simplicity of configuration, in this course we will focus on NFSv4.



Note

The **rpcbind** service replaces **portmap** from Red Hat Enterprise Linux 5.

NFSv4 Demonstration

- Create a user on two machines with a common UID.
- Create a directory to share with NFS and set the proper permissions.
- Edit **/etc/exports**. For example:

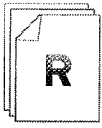
```
/exports/read 192.168.0.0/24(ro, sync)
/exports/write 192.168.0.0/24(rw, sync) 127.0.0.1(rw, sync)
```

- Configure the **nfs** service.

```
[root@serverX]# service nfs start
[root@serverX]# chkconfig nfs on
```

- Mount the NFS pseudo-root share from a client:

```
[root@serverX]# mount -t nfs demo.example.com:/ /mnt
```



References

Red Hat Enterprise Linux Storage Administration Guide

- Unit 10: Network File System (NFS)

exports(5), **exportfs(8)** man pages

Links to NFSv4 Specifications and Resources

<http://www.citi.umich.edu/projects/nfsv4/>



Practice Quiz

NFS Concepts Quiz

1. Under what circumstances should NFSv2 or NFSv3 be used?

2. What is the syntax of the `/etc/exports` file?

3. What steps should be taken to publish a new export on an existing NFSv4 server?

4. Which option tells NFS to allow the root user on client systems to have root privileges in the share as well?

Using NFS

With NFSv4, you can mount the NFS server's pseudo-root export to see what file systems are being exported. If the server supports NFSv3 and earlier, use **showmount -e nfsserver** to talk to **rpc.mountd** and determine what exports are available to which machines.

The **nfs** file system type is used when mounting NFS exports on a client. In Red Hat Enterprise Linux 6, it will try NFSv4 first if supported, then fail back to NFSv3, then NFSv2. To determine the version of NFS in use for a mounted NFS file system, run **mount** with no options or arguments and look for the value of the **vers=** option in the file system's line in the resulting output.

To mount an NFS file system on a client:

- Create an empty directory for the mount point if it does not already exist
- To temporarily mount: **mount -t nfs nfsserver:/export/mount-point**
- To mount immediately at boot, add an appropriate line to **/etc/fstab**:

```
nfsserver:/exports /mount-point nfs defaults 0 0
```

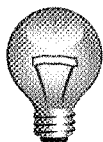


Note

NFS entries in **/etc/fstab** are mounted by the **netfs** service script after networking starts. It is more robust to mount NFS exports on demand with the automounter rather than by using entries in **/etc/fstab** to avoid issues when the client boots if the NFS server is offline or networking is not available.

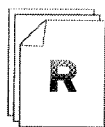
Client-side NFS mount options

- **rw**: mount the file system read-writable
- **ro**: mount the file system read-only
- **vers=4**: try to mount using the NFS version specified only. If this version is not supported by the server the mount request fails
- **soft**: If an NFS request times out, return an error after three retries. Trades off data integrity concerns for increased client responsiveness. (The default behavior is **hard** which will retry indefinitely).



Important

The **intr** mount option is no longer honored in Red Hat Enterprise Linux 6 (by the 2.6.25 Linux kernel and later). Only **SIGKILL (kill -9)** can interrupt a pending NFS operation on newer kernels.

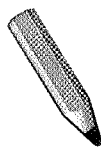


References

Red Hat Enterprise Linux Storage Administration Guide

- Section 10.2: NFS Client Configuration

nfs(5) man page



Test

Criterion Test

Case Study

File Sharing with NFS

Before you begin...

Make sure to run **lab-setup-strickland** from your desktopX system, which will prepare your serverX system for the lab.

Strickland Pro Play is a store specializing in high end recreational equipment and accessories. The new sales software requires requires a file server with two shares that are mounted at each sales station in the store.

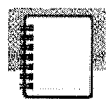
For the file server, deploy an NFSv4 service on desktopX. Create and share two exports on desktopX:

- The first export is for intake of current sales orders. On desktopX, export **/share/current** and make it writable. Root on the client must be able to write to **/share/current** when mounted. The second export is for order archives.
- The second export is to archive old orders. Again on desktopX, export the path **/share/archives** and make it read-only.
- Configure both exports so they are only available to the local classroom network.

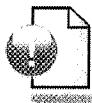
Configure serverX to mount **desktopX:/share/current** as **/sales/current** and **desktopX:/share/archives** as **/sales/archives**. The mounts must be available after a reboot of serverX.

When you are ready, run the **lab-grade-strickland** script on serverX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

NFS Concepts and Configuration

In this section you learned how to:

- Differentiate NFSv4 from older versions of NFS
- Configure an NFS server

Using NFS

In this section you learned how to:

- Export directories from multiple partitions with NFSv4
- Mount NFS filesystems automatically at startup
- Determine appropriate mount options for read-only and read-write access when mounting NFS filesystems as startup



UNIT SEVENTEEN

FILE SHARING WITH CIFS

Introduction

Topics covered in this unit:

- Essential CIFS configuration
- CIFS clients
- Collaborative CIFS shares

Accessing CIFS Shares

CIFS, the Common Internet File System, also known as SMB (Server Message Block), is the standard file and printer sharing system for Microsoft Windows servers and clients. Red Hat Enterprise Linux is able to act as both a client and as a server for CIFS file and printer shares.

In this section, we will review four basic methods for connecting to a CIFS file share:

1. Graphical access to a CIFS share

This technique uses Nautilus to set up an icon for the share which can be used for drag-and-drop access to its contents.

Go to **Places** → **Connect to Server**. Fill in the following fields (leave the others blank and remember to substitute your desktop number for X):

```
Service type: Windows share
Server: serverX
Share: winuserX
User Name: winuserX
Domain Name: CLASSX
```

2. Command-line ftp-style access to a CIFS share: **smbclient**

```
[root@serverX ~]# smbclient -L instructor.example.com
Enter root's password: Enter
Anonymous login successful
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.4-68.el6]
```

Sharename	Type	Comment
-----	----	-----
ftp	Disk	Instructor Public FTP

```
...
[root@serverX ~]# smbclient -L serverX -U winuserX
Enter winuserX's password: winpass
```

Sharename	Type	Comment
-----	----	-----
winuser	Disk	Home Directories

```
...
[root@serverX ~]# smbclient //server1/homes -U winuserX
Enter winuserX's password: winpass
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.5.4-68.el6]
smb: \> ls
.bash_profile                                H          176  Tue Jun 22 11:49:51 2010
```

3. Manually mount a CIFS share

This technique treats the CIFS file share as a standard network file system from a Linux perspective, just like NFS.

```
mount -t cifs -o user=username //server/share /mntpoint
```

Example:

```
[root@serverX ~]# mount -t cifs -o user=winuserX //serverX/winuserX /mnt/
```

4. Persistently mount a CIFS share

This is a variation on the previous example that mounts the CIFS file share automatically at boot time. Note that a credentials file is used in order to provide the username and password for the share.

Add the following line to **/etc/fstab**:

```
//server/share /mntpoint cifs credentials=/etc/filename 0 0
```

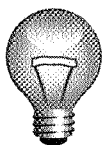
Example:

/etc/fstab:

```
//serverX/homes /mnt/serverX-share cifs credentials=/root/credentials 0 0
```

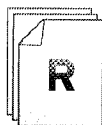
/root/credentials:

```
user=bob  
pass=password
```



Important

The standard Microsoft Windows Uniform Naming Convention is **\\ServerName\Share** to represent a network resource. However, since **** is an escape character rather than a pathname separator in standard Linux shells such as **bash**, the **/** character is normally used in place of **** when expressing a UNC on Linux systems.



References

mount.cifs(8) man page



Practice Quiz

Accessing CIFS Share Quiz

1. What command-line would give ftp-style access to a CIFS share named "common" on a server named "nas2010", logging you in as a user named "winston"?

2. What is wrong with the following line in `/etc/fstab`?

**`\\server\share /mnt/point cifs
user=ralph,pass=password 0 0`**

3. How would you store the login credentials in a separate file to keep them out of **`/etc/fstab`**?

4. When mounting a Windows-based CIFS share, what option allows you to specify the Linux ownership of all mounted files?

Providing Home Directories as CIFS Shares

The Samba service can be used to share Linux file systems as CIFS/SMB network file shares, and Linux printers as CIFS/SMB printer shares. In this section, we will look at the basic configuration of a Samba server and specifically at how to share user home directories using Samba.

CIFS Packages

- **samba-common** - support files for Samba
- **samba-client** - client applications
- **samba** - server applications
- **samba-doc** - documentation (in the "Optional" RHN channel)

Parts of the Samba service

- Packages: *samba-common*, *samba-client*, *samba*, *samba-doc*
- Service script name: **smb**
- Main configuration file: **/etc/samba/smb.conf**

The /etc/samba/smb.conf file

/etc/samba/smb.conf: [global] Section

- **workgroup**

The **workgroup** is used to specify the Windows Workgroup or Domain name for the network.

- **hosts allow**

hosts allow is a comma, space, or tab delimited set of hosts which are permitted to access a service. If specified in the **[global]** section then it will apply to all services, regardless of whether the individual service has a different setting.

You can specify the hosts by name or IP number. For example, you could restrict access to only the hosts on a Class C subnet with something like **allow hosts = 150.203.5**. (using the *trailing dot* notation). The full syntax of the list is described in the man page **hosts_access(5)**.

- **security**

This option affects how clients respond to Samba and is one of the most important settings in the **smb.conf** file.

If your PCs use usernames that are the same as their usernames on the UNIX machine then you will want to use **security = user**. If you mostly use usernames that don't exist on the UNIX box then use **security = share**.

With **security = share**, clients need not log onto the server with a valid username and password before attempting to connect to a shared resource. Instead, the clients send

authentication information on a per-share basis, at the time they attempt to connect to that share.

For **security = user**, the client must log in with a valid user name and password before receiving share information or setting parameters like **guest only**.

security = domain will only work correctly if the machine has been added to the NT Domain. It expects the **encrypted passwords** parameter to be set to **yes**. In this mode Samba will try to validate the username/password by passing it to a Windows NT Primary or Backup Domain Controller, in exactly the same way that a Windows NT Server would do. Note that a valid UNIX user must still exist as well as the account on the Domain Controller to allow Samba to have a valid UNIX account to map file access to. You must set the **password server** parameter to give Samba the server to validate passwords.

Using **security = server**, Samba will try to validate the username/password by passing it to another SMB server. You must set the **password server** parameter to give Samba the server to validate passwords.

For **security = ads**, Samba will act as a domain member in an ADS realm. To operate in this mode, the machine running Samba will need to have Kerberos installed and configured and Samba will need to be joined to the ADS realm using the **net** utility. Read the chapter about **Domain Membership** in the HOWTO for details (this HOWTO is part of the **samba-doc** package). In addition to the **net** command, there is a **netdomjoin-gui** command provided by the **samba-domainjoin-gui** package in the Optional repository for joining a machine to ADS.

/etc/samba/smb.conf: Other Sections

- **[homes]**

This share, which is enabled by default, is a special share that makes user's home directories available via CIFS. It includes **browseable = no**, so it will not show up as an available share until the user authenticates. The share name can either be specified as **homes** (in which case the Samba server will convert it to the home directory path of the user), or **username**.

- **[printers]**

Also available by default, this will share out printers that are currently available.

- **[share]**

If you want to make other shares, place the share name in brackets as above. The share requires at least a **path** parameter. There are several examples in the **smb.conf** file.

Example minimal **/etc/samba/smb.conf** file:

```
[global]
  workgroup = MYGROUP
  server string = Samba Server version %v
  log file = /var/log/samba/log.%m
  max log size = 50

  security = user
  passdb backend = tdbsam
```



```

load printers = yes
cups options = raw

[homes]
    comment = Home Directories
    browsable = no
    writable = yes

[printers]
    comment = All Printers
    path = /var/spool/samba
    browsable = no
    guest ok = no
    writable = no
    printable = yes

```

Other configuration

Samba-only users

- **useradd**

security = user requires a UNIX and Samba account information. Either add a user (preferably using the same name as the Samba account), or place an entry in **/etc/samba/smbusers** (it has some examples). If you are creating a Samba-only user, set the UNIX password to **/sbin/nologin**.

```
[root@serverX ~]# useradd -s /sbin/nologin winuser
```

- **smbpasswd**

If you do not have a Samba password server, you must create authentication data on the local machine. Use **smbpasswd** to create Samba accounts and passwords.

If **smbpasswd** is passed a username without any options, it will attempt to *change* the account password. Passing the **-a** option will add the account and set the password.

```

[root@serverX ~]# smbpasswd -a winuser
New SMB password: winpass
Retype new SMB password: winpass
...
Added user winuser.

```

Securing Samba

- **samba_enable_home_dirs** and **use_samba_home_dirs** SELinux booleans

The **samba_enable_home_dirs** boolean allows local Linux home directories to be exported as CIFS file shares to other systems. The **use_samba_home_dirs** boolean, on the other hand, allows remote CIFS file shares to be mounted and used as local Linux home directories. It is easy to confuse the two options. See the **samba_selinux(8)** man page for more information.

```
[root@serverX ~]# setsebool -P samba_enable_home_dirs on
```

- Service Ports

Samba normally uses TCP/445 for all connections. It also uses UDP/137, UDP/138 and TCP/139 for backward compatibility.



References

smb.conf(5), **smbd**(8), and **samba_selinux**(8) man pages

samba-doc RPM package (in the Optional repository, **/usr/share/doc/samba-doc-*/**)



Practice Performance Checklist

Samba Home Directories Configuration Exercise

Modify the default Samba configuration and security elements to support access to user home directories.

- ☐ Log into serverX and escalate privileges to root
- ☐ Install the necessary package(s) for a Samba server
- ☐ Start and enable the Samba service
- ☐ Configure system to be in the CLASSX workgroup (where X is your station number) with local user definitions.
- ☐ Add a Samba-only user named **winuserX** (where X is your station number) with a Samba password of **winpass**.
- ☐ Enable user home directory access in SELinux
- ☐ Enable the firewall and open up necessary ports to grant access.
- ☐ Test the configuration, by accessing your Samba-only user's home directory from desktopX.

Configuring Group and Print CIFS Shares

Three Steps to Configure a Group Share:

The first step would be to create a collaborative directory in Linux as below:

```
[root@serverX ~]# mkdir -p /shared/dir
[root@serverX ~]# groupadd -r groupname
[root@serverX ~]# chgrp groupname /shared/dir
[root@serverX ~]# chmod 755 /shared
[root@serverX ~]# chmod 2770 /shared/dir
```

Next, we need to set a correct SELinux context on this directory. These steps are needed to set this persistently (across relabels).

```
[root@serverX ~]# semanage fcontext -a -t public_content_t '/shared(/.*)?'
[root@serverX ~]# semanage fcontext -a -t samba_share_t '/shared/dir(/.*)?'
[root@serverX ~]# restorecon -FRvv /shared
```



Note

/shared in the example above is a top-level directory that may be shared using CIFS, NFS, FTP, etc. The **public_content_t** allows each of the services to access the top-level directory. The **/shared/dir** sub-directory is then given **samba_share_t** type that only CIFS can access.

To share this directory via Samba add the following to the bottom of **/etc/samba/smb.conf** and restart the **smb** service.

```
[dir]
path = /shared/dir
valid users = @groupname
writeable = yes
public = no
```

The previous share would only allow users of the group **groupname** to access the share. If you wanted to allow other read-only access to the share, change the Linux permissions on the directory:

```
[root@serverX ~]# chmod 2775 /shared/dir
```

...and change the share in **/etc/samba/smb.conf** to the following:

```
[dir]
path = /shared/dir
writeable = no
write list = @groupname
public = no
```

Two Steps to Individual Printer Share:

To prevent the automatic sharing of all locally defined printers by Samba, remove or comment out the **printers** share in **/etc/samba/smb.conf**:

```
#[printers]
#    comment = All Printers
#    path = /var/spool/samba
#    browseable = no
#    guest ok = no
#    writable = no
#    printable = yes
```

Alternately, change **load printers** from **yes** to **no** (the default is **yes**, so commenting out this line will not work).

To share a particular printer, add the following to **/etc/samba/smb.conf** and restart the service:

```
[myprinter]
comment = My Printer Description
path = /var/spool/samba
read only = yes
printable = yes
printer name = cups_printer_name
```

Limit Client System Access:

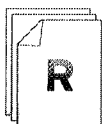
To limit which client systems can access a particular share by IP address, add something like the following to **/etc/samba/smb.conf** in the share section:

```
hosts allow = 192.168.0.1 10.2.12.
```



Note

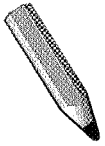
iptables is a non-Samba facility that can limit which client systems can access any CIFS share by IP address.



References

Red Hat Enterprise Linux Deployment Guide

- Section 15.5.1: Group Directories



Test

Criterion Test

Case Study

File Sharing with CIFS

Before you begin...

Make sure to run the **lab-setup-samba** from your desktopX system, which will prepare your serverX system for the lab.

The School of Butler and Hacker has recently deployed several CIFS servers to allow their Windows client systems access to file shares.

The Color Guard, known as Green and Red is deploying a new server and needs to share information using CIFS. That share must be writable by members of the Color Guard, but other people can only have read access.

Enable the firewall and allow all clients on the local network to access the CIFS server.

Configure your serverX to function as a CIFS server, with the following information:

- Workgroup: BUTLER
- Linux Group: greenred
- CIFS Share Name: school
- Directory: /shared/school
- No printers shared

Test the configuration by:

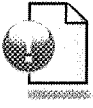
- Creating a user as a member of **greenred** and ensuring they can write to the CIFS share, **school**
- instructor.example.com provides several printers that CUPS should automatically enable. Before you fulfill the printer requirement, check to verify they are available (they should be named printerX). Configure Samba so that no printers are shared and confirm that the user can NOT see them listed with **smbclient**
- Creating a second user not as a member of **greenred** and ensuring they can only read from the CIFS share, **school**

When you are ready, run the **lab-grade-samba** script on serverX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

Accessing CIFS Shares

In this section you learned how to:

- Access CIFS-based file and print shares

Providing Home Directories as CIFS Shares

In this section you learned how to:

- Configure a CIFS based home directory server
- Add Samba-only users to the system

Configuring Group and Print CIFS Shares

In this section you learned how to:

- Create and configure a CIFS share which can be used for group collaboration



UNIT EIGHTEEN

FILE SHARING WITH FTP

Introduction

Topics covered in this unit:

- Safely implement anonymous file sharing with FTP utilizing a "drop-box" for upload

FTP Drop-box Anonymous Upload

FTP, the File Transfer Protocol, is one of the oldest network protocols still in use on the Internet. Many organizations still use FTP for basic file transfers that do not require strong security.

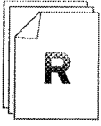
It can be useful to set up a FTP server that allows anonymous users to upload content. However, it is important to configure the server *not* to allow the anonymous users to download content from the upload directory. If the anonymous FTP user can download content from the FTP site that some other anonymous FTP user uploaded, your FTP site can be used by people as a way to transfer illegal or inappropriate content without oversight. It is good practice to ensure that any content served by your site has been approved in some way by an authorized administrator.

In this section, we will look at how to configure an FTP upload directory from which the anonymous user can not download or even list the contents.

1. Create upload directory
 - Group ownership: **ftp**
 - Permissions: group **ftp** has write and execute, but not read access; "other" has no access
2. Modify SELinux for anonymous upload
 - File/directory type context: **public_content_rw_t**
 - Boolean: **allow_ftp_anon_write** must be enabled
3. Modify **/etc/vsftpd/vsftpd.conf**
 - **anon_upload_enable = YES**
 - **chown_uploads = YES**
 - **chown_username = daemon**
 - **anon_umask = 077** (default value)
4. Modify **iptables** to support inbound ftp connections
 - **/etc/sysconfig/iptables_config** change:

```
IPTABLES_MODULES="nf_conntrack_ftp nf_nat_ftp"
```
 - Open new connections to TCP port 21 and allow ESTABLISHED and RELATED network traffic:

```
# iptables -A INPUT -p tcp --dport 21 -j ALLOW
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ALLOW
```



References

Red Hat Enterprise Linux Security Guide

- Section 2.2.6: Securing FTP

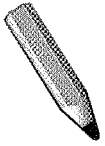
Red Hat Enterprise Linux SELinux Guide

- Section 5.6: Booleans

Red Hat Enterprise Linux Managing Confined Services

- Section 5.4.1: Uploading to an FTP site

ftpd_selinux(8) and **vsftpd.conf**(5) man pages



Test

Criterion Test

Case Study

FTP Drop Box

Before you begin...

Make sure to run the **lab-setup-dropbox** from your desktopX system, which will prepare your serverX system for the lab.

The Quiet Pleases company, a manufacturer of silence cones and other noise canceling devices, has a program to collect information about noise levels around the world. Volunteers have been collecting data about noise and need an easy way to send in reports.

The company has decided to use an FTP server with an anonymous upload directory to collect the reports.

Deploy vsftpd on your serverX and configure a write-only upload directory that is accessible at: `ftp://serverX.example.com/dropbox`

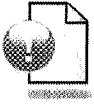
As the volunteers are located all over world, the FTP server must accept connections from anywhere on the internet.

When you are ready, run the **lab-grade-dropbox** script on desktopX to check your work.

How would you address the case study described above? Take notes on your process in the space below and then implement it.



Personal Notes



Unit Summary

FTP Drop-box Anonymous Upload

In this section you learned how to:

- Configure FTP drop-box service
- Manage SELinux to support FTP uploads
- Manage the firewall to support FTP transfers



UNIT NINETEEN

CUPS SERVICE

Introduction

Topics covered in this unit:

- Configuring print queues for a locally connected printer and a printer shared by another system
- Sharing one of your system's print queues with other servers
- Assigning one of your print queues as the "default" print queue
- Disabling or enable a print queue so that it will receive print requests
- Submitting jobs to a print queue
- Listing jobs in a print queue which are waiting to be printed
- Removing a job from a print queue

Configure Printers

The printing system in Red Hat Enterprise Linux is very flexible. Printers may be parallel, serial, or networked. Support is included for printing to remote CUPS IPP, lpd (common Linux and Unix printing subsystem), Windows, Netware, and JetDirect printers.

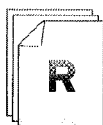
One or more queues is associated with each printer. Print jobs are sent to a queue, not to a printer. Different queues for the same printer may have differing priority or output options. Setting up print queues is the responsibility of the system administrator; individual users do not create print queues.

Printing on a Red Hat Enterprise Linux system is handled by the Common Unix Printing System, or CUPS. The default CUPS installation supports thousands of different printer models, which can be attached to the system locally or over the network. Supported network printer connections include other CUPS servers, older Unix print servers, JetDirect printers and printers shared by Microsoft Windows servers.

A graphical configuration tool is provided to make adding new printers to your system easy. To run this tool, select **System** → **Administration** → **Printing** and follow the instructions to specify your printer's name, manufacturer, model and connection type. Once the printer has been created, it can be selected from the list of printers for further configuration. This interface can also be used to print test pages and set the printer as your system's default.

Configure Printers Demonstration

- Create a new "Generic-text-only" printer.
- Share the print queue.
- Create a network printer that is a raw print queue sent to the local printer above.
- Assign a default printer.
- Enable/disable the printer.



References

`cupsenable(8)`, `cupsdisable(8)` man pages



Practice Group Exercise

Manage Print Queues

1. Create a local print queue and share it with other systems. Name the print queue **local** and make it a text-only printer that points to either the serial or parallel port on your system.



Note

A text-only printer will not accept PostScript files like that sent by the **Print Test** feature. Do not be alarmed that the test page will not print.

2. Create a second print queue that points to a partner's local print queue. Name the print queue **remote** and make it a raw print queue that forwards jobs to your partner's **local** print queue.
3. When you finish, print some text files to **local** and **remote** to verify.



Note

If you are using a serial port, the print jobs are sent to the serial port almost immediately, so it may be difficult to verify that your print queues are working properly. If this is the case, use the "count" files (e.g., c00001, c00002, etc.) in **/var/spool/cups/** to verify. There will be a new count file created every time a print job goes through the queue.

Manage Print Jobs

Once a file has been sent to a queue for printing, it is called a job. Jobs may be canceled when they are in the queue, waiting to be printed.

Manage Print Jobs Demonstration

- *Disable the print queue:* In the GUI, right-click on the printer and uncheck the **Enabled** checkbox. From the CLI, run **cupsdisable *PRINTER***, where *PRINTER* is the name of the printer.
- *Submit a print job:* In a GUI application, use **Ctrl+P** or click on the **Print** button. In the CLI, use the **lpr** or **lp** command.
- *View the queued jobs:* In the GUI, double-click on the printer icon. In the CLI, use the **lpq** or **lpstat** command.
- *Select the pending job and remove it:* In the GUI, open the printer queue, right-click on the printer and choose **Cancel**. In the CLI, find the job ID in the queue and use **lprm** or **cancel** to remove the job.
- *Enable the print queue:* In the GUI, right-click on the printer and check the **Enabled** checkbox. From the CLI, run **cupsenable *PRINTER***, where *PRINTER* is the name of the printer.



References

CUPS Online Help: "Command-Line Printing and Options"
<http://localhost:631/help/options.html>

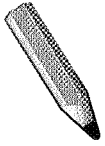
lpr(1), **lpq(1)**, **lprm(1)**, **lp(1)**, **lpstat(1)**, **cancel(1)**, **cupsenable(8)**, **cupsdisable(8)** man pages



Practice Performance Checklist

Print Job Management

- ☐ Disable the default print queue on your system.
- ☐ Submit a print job to the print queue.
- ☐ List the jobs in the default print queue.
- ☐ Cancel the print job you just submitted.
- ☐ Enable the default print queue



Test

Criterion Test

Exercise

Configure and Manage a Printer

Before you begin...

From desktopX, run **lab-setup-cups** to reset your virtual server for this lab.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Configure a network printer to send print jobs to an IPP print queue on instructor.example.com called /printers/printerX where X is your desktop number.
2. Your print queue should be called **remote-test** and should be the default print queue.
3. When you finish, run the evaluation script, **lab-grade-cups**.



Personal Notes



Unit Summary

Configure Printers

In this section you learned how to:

- Configure a printer
- Share a printer

Manage Print Jobs

In this section you learned how to:

- Manage jobs in a print queue
- Enable and disable print queues
- Cancel print jobs



UNIT TWENTY

SSH SERVICE

Introduction

Topics covered in this unit:

- Using SSH keys for login authentication

Using SSH keys

The Secure Shell, **ssh**, allows you to authenticate using a private-public key scheme. This means that you generate two keys, called your private key and your public key. The private key should, as the name implies, be kept private. The public key can be given to anyone. An ssh server that has your public key can issue a challenge that can only be answered by a system holding your private key. As a result, you can authenticate using the presence of your key. This allows you to access systems in a way that does not require typing a password every time but is still secure.

Key generation is done using the **ssh-keygen** command. You can use a key type of DSA or RSA with SSH version 2. SSH protocol version 1 is known to have a security flaw, and therefore its use is not recommended unless you need to connect to legacy ssh servers.

During key generation, you will be given the option to specify a passphrase, which must be provided in order to access your private key. This way, even if the key is stolen, it is very difficult for someone other than you to use it. This gives you time to make a new key pair and remove all references to the old ones, before the private key can be used by an attacker who has cracked it.

It is always wise to passphrase-protect your private key since the key allows you to access other machines. However, this means that you must type your passphrase whenever the key is used, making the authentication process no longer password-less. This can be avoided using **ssh-agent**, which can be given your passphrase once at the start of your session (using **ssh-add**) so it can provide it as necessary while you stay logged in.

Once your SSH keys have been generated, they are stored by default in the **.ssh/** directory of your home directory. Default modes should be 600 on your private key and 644 on your public key.

Before you can use key-based authentication, you will need to copy your public key to the destination system. This can be done with **ssh-copy-id**.

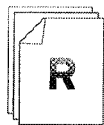
```
[student@desktopX ~]$ ssh-copy-id -i .ssh/id_rsa.pub root@desktopY
```

When you copy your key to another system via **ssh-copy-id**, it uses the **~/.ssh/id_rsa.pub** file by default. If you use a different key, or give your key a different name, it will have to be specified with the **-i** option when using **ssh-copy-id**.

SSH Key Demonstration

- Use **ssh-keygen** to create a public-private keypair.
- Use **ssh-copy-id** to copy the public key to the correct location on a remote system. For example:

```
[root@serverX]# ssh-copy-id root@serverY.example.com
```

References

Red Hat Enterprise Linux Deployment Guide

- Section 9.2.4: Using a Key-Based Authentication

ssh-keygen(1), **ssh-copy-id**(1), **ssh-agent**(1), **ssh-add**(1) man pages

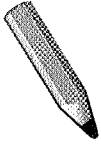


Practice Exercise

Using SSH Keys

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Create an SSH key pair as **student** on desktopX.
2. Install the SSH public key for the **student** account on serverX.
3. Connect to serverX from desktopX using the SSH keys.



Test

Criterion Test

Exercise

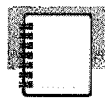
Securing SSH

Before you begin...

Run the **lab-setup-server** command as **root** on your desktopX system. This will prepare your serverX system for the lab.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Copy the SSH public key generated previously on desktopX to the **student** account on serverX.
2. Confirm you can **ssh** into serverX as **student** from desktopX using the SSH keys.



Personal Notes



Unit Summary

Using SSH keys

In this section you learned how to:

- Create and use SSH keys



UNIT TWENTY ONE

VIRTUAL NETWORK COMPUTING (VNC) SERVICE

Introduction

Topics covered in this unit:

- Configuring a remote desktop
- Connecting to a VNC server securely

Configuring a VNC Server

While many data centers will standardize on **ssh** for remote administration of Unix and Linux systems, some will use Virtual Network Computing (VNC) for remote administration of Windows servers. Red Hat Enterprise Linux 6 supports the implementation of a VNC server that can allow one or more remote graphical desktops.

Configure a VNC server demonstration

1. Install the VNC server package

```
[root@demo ~]# yum install tigervnc-server
```

2. Edit `/etc/sysconfig/vncservers`:

```
VNCSERVERS="2:root"
VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
```

The **-localhost** option will prevent remote VNC clients connecting except when doing so through a secure tunnel, for example, when using **vncviewer** and its **-via** option:

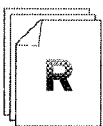
```
vncviewer -via user@remotehost localhost:2
```

3. Set a VNC password.

```
[root@demo ~]# vncpasswd
Password: password
Verify: password
```

4. Start and enable the service:

```
[root@demo ~]# service vncserver start
[root@demo ~]# chkconfig vncserver on
```



References

Red Hat Enterprise Linux Deployment Guide
• Section 18.1.23 - `/etc/sysconfig/vncservers`

vncviewer(1), **vncpasswd**(1) man pages



Practice Exercise

Enabling a VNC Server

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Install the **tigervnc-server** package on serverX.
2. Configure VNC display 1 for student. Add the following to **/etc/sysconfig/vncservers**:

```
VNCSERVERS="1:student"
```

3. Set **redhat** as the VNC password for student:

```
[student@serverX ~] vncpasswd  
Password: redhat  
Verify: redhat
```

4. Start and enable the VNC service.
5. You will verify the connection in the next section.

Secure access to a remote GNOME desktop

The **vncviewer** command is a viewer (client) used to connect to a VNC server running on a remote system.



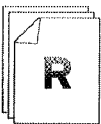
Warning

Use the **-via** option to tunnel VNC traffic over an SSH tunnel whenever possible. VNC is a cleartext protocol and your passwords and desktop session will be vulnerable to eavesdropping and interference if you do not tunnel it over a secure connection.

Connect to VNC server securely demonstration

1. Connect to a VNC server using SSH:

```
[root@instructor ~]# vncviewer -via visitor@demo localhost:1
```



References

vncviewer(1) man page



Practice Exercise

Connect to VNC securely

Carefully perform the following steps. Ask your instructor if you have problems or questions.

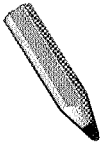
1. Configure the VNC server on serverX to allow local connections only. Edit **/etc/sysconfig/vncservers** and add the following:

```
VNCSERVERARGS[1]="-localhost"
```

2. Connect to the VNC server on serverX securely from desktopX using an SSH tunnel:

```
[student@desktopX ~] vncviewer -via serverX localhost:1
```

3. Verify everything is completed as specified.



Test

Criterion Test

Exercise

Configure Multiple Desktops with VNC

Before you begin...

Run the **lab-setup-server** command as **root** on your desktopX system. This will prepare your serverX system for the lab.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

1. Install the VNC server package on serverX.
2. Configure display 1 for **student** and display 2 for **visitor**.
3. Only permit connections from localhost.
4. Set **redhat** as the VNC passwords for both **student** and **visitor**.
5. Start and enable the VNC service.
6. Verify everything is completed as specified, then check your work using a secure connection.



Note

The parameter after **-via** is used to connect using **ssh**. It is not necessary to use the username to whose VNC session you are connecting. Any username would work, as long as you know the password.



Personal Notes



Unit Summary

Configuring a VNC Server

In this section you learned how to:

- Configure a VNC server

Secure access to a remote GNOME desktop

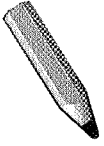
In this section you learned how to:

- Connect securely to a VNC server



UNIT TWENTY TWO

COMPREHENSIVE REVIEW



Test

Comprehensive Review Test

Exercise

Comprehensive Review

Before you begin...

Run the **lab-setup-server** command as **root** on desktopX.

Carefully perform the following steps. Ask your instructor if you have problems or questions.

Configure serverX so that it meets the following requirements. For all services, allow connections from the local 192.168.0.0/24 subnet, but disallow connections from the 192.168.1.0/255.255.255.0 subnet.

1. Configure SELinux to run in Enforcing mode.
2. Allow SSH connections from the local subnet.
3. Configure an SMTP server that allows connections from the local subnet.
4. Connect to the LDAP server, `instructor.example.com`, using the distinguished name (DN) of **dc=example, dc=com** for account information. The LDAP server requires secure connections using the certificate found at `ftp://instructor.example.com/pub/EXAMPLE-CA-CERT`. The LDAP server provides an account named **ldapuserX**.

Use Kerberos passwords with a realm **EXAMPLE.COM** for authentication. Set the KDC and Admin servers to `instructor.example.com`. The accounts have a password of **kerberos**.
5. Configure an automounted home directory for the **ldapuserX** account. The home directory is shared via NFS from `instructor.example.com`.
6. Connect to the iSCSI target **rdisks.serverX** provided by `instructor.example.com`.
7. Remove all of the current partitions on the iSCSI disk. Configure a new 30 MB physical partition using the iSCSI target with an ext4 filesystem and a label of **test** mounted on **/test/**. The **/test/** directory must be owned by the user **root** and the group **root**, and have a permission of 755.
8. Configure a new 1 GB logical volume named **mylv** in the **vgsrv** volume group, with an ext4 filesystem mounted on **/mylv/**.
9. Configure NFS to share the **/test/** directory. Make it read-only to the local subnet. Allow **root** to have root privileges when accessing the NFS share.
10. Create a user account named **matt** using a password of **matt**.
11. Create a user account named **cindy** using a password of **cindy**.
12. Create a group named **admins** that includes **matt** and **cindy**.

-
13. Configure Samba to share the **/test/** directory using a share name of **test**. Make it readable for **cindy** (use a Samba password of **password**) and writable for **matt** (use a Samba password of **password**). Make sure the Linux permissions allow read/write as listed here, as well as meeting the user, group and permission requirements listed above.

14. Configure a secure web server using the certificate and key located at *http://instructor/pub/materials/tls/certs/serverX.crt* and *http://instructor/pub/materials/tls/private/serverX.key*. Make the web server use **/mylv/index.html** as the default web page. Configure the **index.html** file such that accessing the secure web site will present the following:

Hello World!

15. Allow **cindy** and **matt** to write the **/mylv/index.html** file.



Personal Notes

Appendix A. Solutions

Software Management



Practice Quiz

Red Hat Network Registration

1. The menu item that begins the registration with Red Hat Network is System → Administration → RHN Registration.
2. The first registration choice determines whether a system registers with Hosted RHN or RHN Satellite.
3. Optionally additional web proxy server and authentication information may need to be provided.
4. An RHN user name or RHN account and its matching password must be provided for successful Red Hat Network registration.
5. The last questions to be answered during the registration process are system name and whether to upload hardware and software or package profile information.



Practice Exercise

Using YUM repositories

You will configure your server to use a separate YUM repository to obtain updates, and update your machine.

1. Create the file `/etc/yum.repos.d/errata.repo`, to enable the "Updates" repository found on the instructor machine. It should access content found at the following URL: `ftp://instructor.example.com/pub/rhel6/Errata`

Create the file `/etc/yum.repos.d/updates.repo` with the following content:

```
[updates]
name=Red Hat Updates
baseurl=ftp://instructor.example.com/pub/rhel6/Errata
enabled=1
gpgcheck=1
```

2. Update all relevant software provided by the repository, using **yum update**.

yum update



Practice Exercise

Searching for and installing packages

Login as **root** on serverX and perform the following tasks:

1. Attempt to run the command **gnuplot**. You should find that it is not installed.
2. Search for plotting packages.

```
yum search plot
```

3. Find out more information about the **gnuplot** package.

```
yum info gnuplot
```

4. Install the **gnuplot** package.

```
yum install gnuplot
```

5. Attempt to remove the **gnuplot** package, but say no.

```
yum remove gnuplot
```

How many packages would be removed? 1

6. Attempt to remove the **gnuplot-common** package, but say no.

```
yum remove gnuplot-common
```

How many packages would be removed? 2



Practice Exercise

Handling Third-Party Software

In this exercise you will gather information about a third-party package, extract files from it, and install it as a whole on your desktopX system.

1. Download *wonderwidgets-1.0-4.x86_64.rpm* from <http://instructor/pub/materials>.
2. What files does it contain?

```
rpm -qlp wonderwidgets-1.0-4.x86_64.rpm
```

3. What scripts does it contain?

```
rpm -qp --scripts wonderwidgets-1.0-4.x86_64.rpm
```

4. How much disk space will it use when installed?

```
rpm -qip wonderwidgets-1.0-4.x86_64.rpm
```

5. Use **yum localinstall** to install the package.

```
yum localinstall wonderwidgets-1.0-4.x86_64.rpm
```



Practice Quiz

RPM Spec File

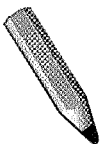
1. The package **Version** is usually derived from the open source project while the package **Release** is the packager's version.
2. The **Group** directive categorizes the type of package being built.
3. The name of the tarball containing the files used to build the package is specified with the **Source** directive.
4. The **BuildArch** directive specifies the target architecture the package is being built for. **noarch** will be its value when the package can be installed on any architecture.
5. The **Summary** directive specifies the 1-line description of a package while the **%description** section provides a more thorough explanation of what that package is for.
6. The **%install** section contains the code used to place files in the **\$RPM_BUILD_ROOT** chroot directory structure.
7. The **%files** section defines which files and directories to package into the RPM.
8. The **%prep**, **%build**, and **%clean** sections contain shell code used to assemble a package and clean up after it has been built.



Practice Quiz

Create a Yum Repository Quiz

1. Install the **createrepo** package if necessary.
2. Create a directory that can be **shared (via FTP or HTTP)**.
3. Create a subdirectory called **Packages**.
4. Copy **all RPM packages** to be published into **Packages**.
5. Execute **createrepo** on the **top-level** directory.



Test

Criterion Test

Performance Checklist

Create an RPM

- Download the the file `ftp://instructor.example.com/pub/materials/hello.sh`.

```
[student@serverX ~]$ mkdir ~/hello-1.0
[student@serverX ~]$ cd ~/hello-1.0
[student@serverX hello-1.0]$ wget ftp://instructor.example.com/pub/materials/hello.sh
```

- Create a simple RPM that installs **hello.sh** in **/root/bin**. Make sure that **hello.sh** is installed with a mode of 755.

```
[student@serverX hello-1.0]$ cd
[student@serverX ~]$ mkdir -p ~/rpmbuild/SOURCES
[student@serverX ~]$ mkdir -p ~/rpmbuild/SPECS
[student@serverX ~]$ tar -cvzf ~/rpmbuild/SOURCES/hello-1.0-1.tar.gz hello-1.0
```

`~/rpmbuild/SPECS/hello.spec` should look like:

```
Name:      hello
Version:    1.0
Release:    1
Summary:    Hello
Group:      RHCE
License:    GPL
URL:        http://www.redhat.com
Source0:    %{name}-%{version}-%{release}.tar.gz
BuildRoot:  /var/tmp/%{name}-buildroot

%description
Installs /root/bin/hello.sh

%prep
%setup -q -n %{name}-%{version}

%build

%install
rm -rf $RPM_BUILD_ROOT
mkdir -p $RPM_BUILD_ROOT/root/bin
install -m 755 hello.sh $RPM_BUILD_ROOT/root/bin/hello.sh

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root,-)
/root/bin/hello.sh

%changelog
```

```
[student@serverX ~]$ su -
Password: redhat
[root@serverX ~]# yum install -y rpm-build
[root@serverX ~]# exit
[student@serverX ~]$ rpmbuild -ba ~/rpmbuild/SPECS/hello.spec
```

- Create a GPG key and sign the package with the key. Export the public GPG key.



Note

You must have a graphical session available to successfully generate a GPG key. **gpg** now uses a graphical application to enter and validate the key.

```
[student@serverX ~]$ gpg --gen-key
gpg (GnuPG) 2.0.14; Copyright (C) 2009 Free Software Foundation, Inc. This is free
software: you are free to change and redistribute it. There is NO WARRANTY, to
the extent permitted by law. Please select what kind of key you want: (1) RSA and
RSA (default) (2) DSA and Elgamal (3) DSA (sign only) (4) RSA (sign only) Your
selection? Enter
RSA keys may be between 1024 and 4096 bits long. What keysize do you want?
(2048) Enter
Requested keysize is 2048 bits Please specify how long the key should be valid. 0
= key does not expire <n> = key expires in n days <n>w = key expires in n weeks
<n>m = key expires in n months <n>y = key expires in n years Key is valid for?
(0) Enter
Key does not expire at all Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key. Real name: My Name
Email address: student@serverX.example.com
Comment: Enter
You selected this USER-ID: "My Name <student@serverX.example.com>" Change (N)ame,
(C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key. Enter passphrase
Passphrase: testing123
Please re-enter this passphrase. Passphrase: testing123
We need to generate a lot of random bytes. It is a good idea to perform some other
action (type on the keyboard, move the mouse, utilize the disks) during the prime
generation; this gives the random number generator a better chance to gain enough
entropy.

gpg: /home/student/.gnupg/trustdb.gpg: trustdb created
gpg: key 54AF5285 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/54AF5285 2010-12-09
Key fingerprint = 315F E90B 1745 2288 EBAE 4E7B 4BC6 4568 54AF 5285
uid My Name <student@serverX.example.com>
sub 2048R/D08B2951 2010-12-09
```

To export the key, find the key ID from the output above. It can be found after the **pub 2048R/** output above. In this example, the key ID is **54AF5285**. The following examples will show the commands using this key ID. Your key ID will be different, so replace the example key ID with your key ID.

```
[student@serverX ~]$ gpg -a -o ~/RPM-GPG-KEY-student --export 54AF5285
```

Create the **~/rpmmacros** file and add the following content:

```
%_pgp_name 54AF5285
```

Sign the RPM package

```
[student@serverX ~]$ rpm --resign ~/rpmbuild/RPMS/x86_64/hello-1.0-1.x86_64.rpm
Enter pass phrase: testing123
Pass phrase is good. /home/instructor/rpmbuild/RPMS/x86_64/hello-1.0-1.x86_64.rpm:
```

- ☐ Deploy a web server and create a yum repository in **/var/www/html/Packages/**. Create a repository file that references **http://serverX/Packages**. Serve the GPG key from the web server and include the key in the repository file.

```
[root@serverX ~]# mkdir /var/www/html/Packages
[root@serverX ~]# cp ~student/rpmbuild/RPMS/x86_64/hello-1.0*.rpm /var/www/html/
Packages/
[root@serverX ~]# cp ~student/RPM-GPG-KEY-student /var/www/html/Packages/
[root@serverX ~]# createrepo -v /var/www/html/Packages/
[root@serverX ~]# service httpd start
```

Create the **/etc/yum.repos.d/hello.repo** file with the following content:

```
[hello]
name=hello
description=ServerX Yum Repo
baseurl=http://serverX.example.com/Packages
enabled=1
gpgcheck=1
gpgkey=http://serverX.example.com/Packages/RPM-GPG-KEY-student
```

- ☐ Install your rpm using the yum repository above and run **/root/bin/hello.sh**.

```
[root@serverX ~]# yum -y install hello
[root@serverX ~]# hello.sh
```


Network Management



Practice Quiz

Advanced Network Interface Configuration Quiz

1. Which mode of Linux Ethernet bonding primarily uses one slave interface and changes interface upon failure?

(select one of the following...)

- a. Mode 0 (balance-rr)
- b. Mode 1 (active-backup)
- c. Mode 3 (broadcast)

2. Which mode of Linux Ethernet bonding uses all interfaces in a round robin fashion to achieve more throughput?

(select one of the following...)

- a. Mode 0 (balance-rr)
- b. Mode 1 (active-backup)
- c. Mode 3 (broadcast)

3. When creating a bonded network interface, which configuration file contains the IP address and netmask definitions for the interface?

(select one of the following...)

- a. **/etc/sysconfig/network**
- b. **/etc/sysconfig/network-scripts/ifcfg-bond0**
- c. **/etc/sysconfig/network-scripts/ifcfg-iface**
- d. None of the above

4. When creating a bonded network interface, which configuration file defines the type of bonding?

(select one of the following...)

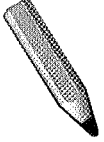
- a. **/etc/sysconfig/network**
- b. **/etc/sysconfig/network-scripts/ifcfg-bond0**
- c. **/etc/sysconfig/network-scripts/ifcfg-iface**
- d. None of the above

5. When creating a bonded network interface, which variable definitions must be specified in the **/etc/sysconfig/network-scripts/ifcfg-iface** configuration file?

(select one of the following...)

- a. **GATEWAY**
- b. **IPADDR**
- c. **MASTER**

- d. None of the above



Test

Criterion Test 1

Case Study

Routing Network Traffic: Operation Strategic Holistic Unusual

Before you begin...

Run the **lab-setup-oshu** script on desktopX.

Operation Strategic Holistic Unusual (or OSHU), is an online chat system for fans of conspiracy fiction. In order to join the site they have two requirements, listed below.

1. To fulfill the first requirement, you must prove your ability to “disappear” a server. You will do this by modifying the configuration on serverX so that it does not respond to any ping requests. Make this change persistent so that it will still be in effect after a reboot.
2. The second requirement is to join the “secret” OSHU network. To join the network, add an additional IP address to serverX, where X is your desktop/server number:

10.42.10.X/24

When you have fulfilled the requirements, run **lab-grade-oshu** on desktopX to check your work.

1. Add the following to **/etc/sysctl.conf**

```
net.ipv4.icmp_echo_ignore_all = 1
```

2. Enable the setting

```
[root@serverX ~] sysctl -p
```

3. Configure static network settings

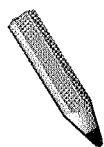
```
[root@serverX ~] service NetworkManager stop  
[root@serverX ~] chkconfig NetworkManager off
```

4. Create the **/etc/sysconfig/network-scripts/ifcfg-eth0:0** file and add the following content:

```
DEVICE=eth0:0  
IPADDR=10.42.10.X  
NETMASK=255.255.255.0  
ONPARENT=yes
```

5. Enable the new network settings

```
[root@serverX ~] ifup eth0:0
```



Test

Criterion Test 2

Exercise

Troubleshooting Network Configuration from the command-line

All of the following should be performed on your virtual server, serverX. You will start by running a script that will "break" your network configuration. You will have five minutes to resolve each of the two problems. Be sure to document what you have found, as we will review at the end.

The following are your network settings for serverX:

IP Address: 192.168.0.X+100
Netmask: 255.255.255.0 (/24)
DNS Server: 192.168.0.254
Default Gateway: 192.168.0.254

1. Run the first script to misconfigure your networking:

lab-break-net 1

2. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
3. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
4. Document what you have found

Use this space for notes

In this first problem, the hostname `instructor.remote.test` was not being resolved to an IP address. `/etc/resolv.conf` was pointing to the wrong DNS server. Correcting the problem by modifying `/etc/sysconfig/network-scripts/ifcfg-eth0` to have **DNS1=192.168.0.254**

5. Run the second script to misconfigure your networking:

lab-break-net 2

6. Symptom: A web browser is unable to access the web page at `http://instructor.remote.test`
7. Apply the three steps: TEST, CHECK, FIX to identify and resolve the problem.
8. Document what you have found

Use this space for notes

In this second problem, the host `instructor.remote.test` is on a separate network and unreachable. **ip route** showed the default gateway pointing to the wrong router. Correcting the problem by modifying `/etc/sysconfig/network-scripts/ifcfg-eth0` to have **GATEWAY=192.168.0.254**

Storage Management



Practice Quiz

Add a New Filesystem

1. Identify a disk that has some free space `fdisk -cul`
2. Create a new partition on that disk `fdisk -cu /dev/device`
3. Update the kernel partition table `reboot`
4. Create a filesystem on the partition `mkfs -t ext4 /dev/device`
5. Add an entry to the filesystem table file Add an entry to `/etc/fstab` like the following:
`UUID=cb79b7d0-dc14-4402-8465-6857346c9a53 /directory ext4 defaults 1 2`
6. Create a mount point `mkdir /directory`
7. Mount the filesystem `mount -a`



Practice Resequencing Exercise

Create Encrypted Filesystem

For each of the file or directory names below, write down the number of its definition from the list at the bottom.

- | | |
|--|---|
| <u>1</u>
<u>4</u>
<u>2</u>
<u>6</u>
<u>8</u>
<u>5</u>
<u>3</u>
<u>7</u>
<u>9</u> | Create a new partition
Create an ext4 filesystem
Format the new partition for encryption
Mount the filesystem on the unlocked device
Create an entry in /etc/fstab
Create a directory to use as a mount point
Unlock the encrypted partition
Create an entry in /etc/crypttab
Make LUKS aware of the password file |
|--|---|
-
1. **fdisk**
 2. **cryptsetup luksFormat /dev/vdaN**
 3. **cryptsetup luksOpen /dev/vdaN secret**
 4. **mkfs -t ext4 /dev/mapper/secret**
 5. **mkdir /secret**
 6. **mount /dev/mapper/secret /secret**
 7. **secret /dev/vdaN /password/file**
 8. **/dev/mapper/secret /secret ext4 defaults 1 2**
 9. **cryptsetup luksAddKey /dev/vdaN /password/file**

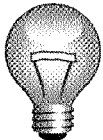


Practice Exercise

Create and use a new swap partition.

Create and use a new 256 MB swap partition on your virtual server, serverX.

1. Start **fdisk** and create a new partition



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand

2. Change the partition type to **swap**.

Type **t** to change the partition type to "0x82 Linux Swap"

3. Prepare the new partition for use as swap

```
mkswap /dev/vdaN
```

(where *N* is the partition number)

4. Determine the UUID

```
blkid /dev/vdaN
```

5. Add the new partition to **/etc/fstab**

```
UUID=uuid swap swap defaults 0 0
```

6. Determine current amount of swap

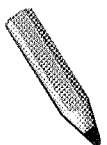
```
swapon -s
```

7. Activate the new swap

```
swapon -a
```

8. Verify newly activated swap

```
swapon -s
```



Test

Criterion Test

Exercise

Partitions and Filesystems Lab

Before you begin...

Reset serverX by running **lab-setup-server** from desktopX.

1. On serverX, connect to the iSCSI target **iqn.2010-09.com.example:rdisks.serverX** from 192.168.0.254 and ensure it is enabled at boot time.

```
[root@serverX ~]# service iscsid start
[root@serverX ~]# chkconfig iscsid on
[root@serverX ~]# iscsiadm -m discovery -t st -p 192.168.0.254
[root@serverX ~]# iscsiadm -m node -T iqn.2010-09.com.example:rdisks.serverX -p
192.168.0.254 -l
```

2. Create two new physical partitions using the iSCSI disk, 10 MB in size each.

```
[root@serverX ~]# fdisk -cu /dev/sda
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x14d0f83d.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
```

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 1

First sector (2048-65535, default 2048): Enter

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-65535, default 65535): +10M

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): Partition number (1-4): 2

First sector (22528-65535, default 22528): Enter

Using default value 22528

Last sector, +sectors or +size{K,M,G} (22528-65535, default 65535): +10M

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

3. With the first partition, create an ext4 filesystem mounted on **/test** and make it persistent.

```
[root@serverX ~]# mkfs -t ext4 /dev/sda1
[root@serverX ~]# blkid /dev/sda1
/dev/sda1: UUID="14ec9746-b443-4f89-af7b-d827adfd3de1" TYPE="ext4"
```

Copy the "UUID=XXXXXXXX" line from the output and add an entry to **/etc/fstab**:

```
UUID=XXXXXXXX swap swap defaults 0 0
```

For the example above it would be:

```
UUID=14ec9746-b443-4f89-af7b-d827adfd3de1 /test ext4 defaults 1 2
```

```
[root@serverX ~]# mkdir /test [root@serverX ~]# mount -a
```

4. With the second partition, create an ext4 file system persistently mounted on **/opt** with **acl** as a default mount option.

```
[root@serverX ~]# mkfs -t ext4 /dev/sda2
[root@serverX ~]# blkid /dev/sda2
/dev/sda2: UUID="c261f7fb-b2e9-4678-b7e4-61293c87d095" TYPE="ext4"
```

Copy the UUID from the output and use this information to add an entry to **/etc/fstab**

```
UUID=XXXXXXXX /opt ext4 acl 1 2
```

For the example above, it should read:

```
UUID=c261f7fb-b2e9-4678-b7e4-61293c87d095 /opt ext4 acl 1 2
```

```
[root@serverX ~]# mount -a
[root@serverX ~]# df -h /opt
Filesystem Size Used Avail Use% Mounted on
/dev/sda2    9.7M  1.1M  8.1M  12% /opt
```


Logical Volume Management

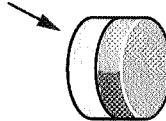


Practice Quiz

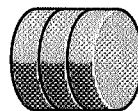
LVM Components

- Fill in the following graphic with the names of the components.

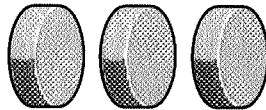
Unused Space



4. _____



3. _____



2. _____



1. _____

1. Physical storage 2. Physical volume(s) 3. Volume group 4. Logical volume(s)

- What are the smallest pieces (chunks or blocks) of the physical volume?
Physical Extents

- What is the smallest size you could make a logical volume?

The size of a single physical extent.

4. What references the physical extents of a logical volume?
Logical extents
-

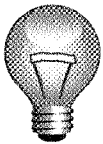


Practice Exercise

Implement LVM and Create a Logical Volume

All of these steps will be performed on serverX.

1. Create a new partition of 512 MB and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

```
[root@serverX ~]# fdisk -cu /dev/vda
n
1      (logical partition)
[enter] (default start)
+512M
t
8      (this partition is /dev/vda8)
8e     (LVM type)
w

[root@serverX ~]# reboot    (to reload the partition table)
[root@serverX ~]# pvcreate /dev/vda8
```

2. Create a Volume Group named **shazam** using the Physical Volume created in the previous step.

```
[root@serverX ~]# vgcreate shazam /dev/vda8
```

3. Create and format with **ext4**, a new Logical Volume of 256 MB called **/dev/shazam/storage**.

```
[root@serverX ~]# lvcreate -n storage -L 256MB shazam
[root@serverX ~]# mkfs -t ext4 /dev/shazam/storage
```

4. Modify your system such that **/dev/shazam/storage** is mounted at boot time as **/storage**.

```
[root@serverX ~]# mkdir /storage
```

Update `/etc/fstab` with the following entry:

```
/dev/shazam/storage    /storage    ext4    defaults    1 2
```

```
[root@serverX ~]# mount -a
```

```
[root@serverX ~]# df
```

```
[root@serverX ~]# reboot    (to confirm persistence)
```



Practice Exercise

Extend a Logical Volume

All of these steps will be performed on `serverX`.

1. Determine the amount of free space in Volume Group **shazam**.

```
[root@serverX ~]# vgdisplay shazam
```

2. Extend the logical volume `/dev/shazam/storage` with *half* the available extents in the volume group using command-line tools.

If there were 100 free extents, the following command would extend the logical volume using half of them:

```
[root@serverX ~]# lvextend -l +50 /dev/shazam/storage
```

3. Extend the filesystem mounted on `/storage` using command-line tools.

```
[root@serverX ~]# resize2fs /dev/shazam/storage
```

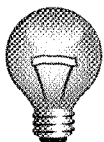


Practice Exercise

Extend a Volume Group

All of these steps will be performed on `serverX`.

1. Create a new 512 MB partition and prepare it for use with LVM as a Physical Volume.



Important

To have room for creating additional partitions in the future, if needed, be sure to create an Extended partition beforehand.

Use **fdisk** and **pvccreate** to prepare the physical volume.

```
[root@serverX ~]# fdisk -cu /dev/vda
n
1      (logical partition)
[enter]  (default start)
+512M
t
9      (this partition is /dev/vda9)
8e     (LVM type)
w

[root@serverX ~]# reboot      (to reload the partition table)
[root@serverX ~]# pvccreate /dev/vda9
```

2. Extend the Volume Group **shazam** by adding the Physical Volume created in the previous step.

Use **vgextend** to extend the volume group:

```
[root@serverX ~]# vgextend shazam /dev/vda9
[root@serverX ~]# vgdisplay shazam      (check size and free space)
```



Practice Exercise

Creating an LVM Snapshot

Compare the contents of our existing logical volume, **/dev/shazam/storage**, to a new snapshot volume, **/dev/shazam/storagesnap**, while making changes to the original volume.

All of these steps will be performed on serverX.

1. Copy the file **/usr/share/dict/linux.words** to **/storage** so you have some data to compare.

```
[root@serverX ~]# cp /usr/share/dict/linux.words /storage
```

2. Create a new 20 MB snapshot logical volume of **/dev/shazam/storage** called **storagesnap**.

```
[root@serverX ~]# lvcreate -n snapstore -L20M -s /dev/shazam/storage
```

3. Manually mount **/dev/shazam/storagesnap** read only at **/storagesnap**

```
[root@serverX ~]# mkdir /storagesnap
[root@serverX ~]# mount -o ro /dev/shazam/storagesnap /storagesnap
```

4. List the contents of **/storagesnap** and note that they are the same as **/storage**.

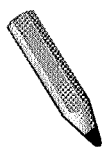
```
[root@serverX ~]# ls /storagesnap ; ls /storage
```

5. Delete the file **/storage/linux.words** and note that it still exists in **/storagesnap**.

```
[root@serverX ~]# rm /storage/linux.words
[root@serverX ~]# ls /storagesnap
```

6. Clean up: unmount **/storagesnap**, remove the directory, and delete the **storagesnap** logical volume.

```
[root@serverX ~]# umount /storagesnap
[root@serverX ~]# rmdir /storagesnap
[root@serverX ~]# lvremove /dev/shazam/storagesnap
```



Test

Criterion Test

Case Study

LVM Case Study

Before you begin...

Make sure to run the **lab-setup-lvm** from your desktopX system, which will prepare your serverX system for the lab.

Allison needs to store data for her business. Her customer database is currently 256 MB in size. The data in the database changes about 10 MB per hour on a typical day. The backup software takes 10 minutes to complete a full run.

Create a new Volume Group called **allison** with enough space for both a 512 MB volume and a snapshot of that volume for the backup software. Create a 512 MB logical volume for Allison's customer database called **custdb**. Create a snapshot volume of Allison's customer database called **custdbsnap** for her backup software.

When you are ready, run the **lab-grade-lvm** script on serverX to check your work.

1. Create a new 1 GB partition using **fdisk** and prepare it for use with LVM

```
[root@serverX ~]# fdisk -cu /dev/vda

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-12582911, default 9914368): Enter
Using default value 9914368
Last sector, +sectors or +size{K,M,G} ((9914368-12582911, default 12582911): +1G
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
```

The kernel still uses the old table. The new table will be used at the next reboot or after you run `partprobe(8)` or `kpartx(8)` Syncing disks.

```
[root@serverX ~]# reboot
```

```
[root@serverX ~]# pvcreate /dev/vda3
Physical volume "/dev/vda3" successfully created
```

2. Create a new volume group called **allison** using the new partition

```
[root@serverX ~]# vgcreate allison /dev/vda3
Volume group "allison" successfully created
```

3. Create a 512 MB logical volume for Allison's customer database

```
[root@serverX ~]# lvcreate -n custdb -L512M allison
Logical volume "custdb" created
```

4. Create a 10 MB snapshot volume of Allison's customer database

```
[root@serverX ~]# lvcreate -n custdbsnap -L10M -s /dev/allison/custdb
Rounding up size to full physical extent 12.00 MiB
Logical volume "custdbsnap" created
```

Account Management



Practice Performance Checklist

Managing Password Aging Policies

Your instructor will divide you into small groups. Within each group, discuss which password aging policies would be appropriate for *professors* (who will be using the machine for a long time), *graduate students* (who will be using the machine for a few years), and *summer interns* (who will only be using the machine for the summer).

- **Professors:** faraday, juliet
- **Graduate Students:** jack, kate, james
- **Summer Interns:** walt, ben, clair, hugo

- ☐ If you do not already have the users and groups defined, run **lab-add-users** on serverX.

```
[root@serverX ~]# lab-add-users
```

- ☐ For each group of users, determine a password aging policy which would be appropriate, including
 - Account expiration dates (if appropriate).
 - Time before passwords must be changed.
 - Time before unchanged passwords force an account to go inactive.

Professor accounts will not expire. They must change passwords once a quarter (every 90 days). Once a password has expired, their account will become inactive after 30 days.

Graduate student accounts will not expire. They must change passwords once a month (every 30 days). Once a password has expired, their account will become inactive after 30 days.

Summer intern accounts expire at the end of summer (our example will assume the summer of 2011). They must change their password once a month (every 30 days). Once a password has expired, their account will become inactive after 7 days.

- ☐ Once determined, use **chage** to implement your policy for the users added in the previous section, according to their role.

Additionally, force all users to change their password on first login.

The following is the adjustments made to faraday's (a professor) account.

```
[root@serverX ~]# chage -M 90 -I 30 faraday
```

The following is the adjustments made to kate's (a grad student) account.

```
[root@serverX ~]# chage -M 30 -I 30 kate
```

The following is the adjustments made to hugo's (a summer intern) account.

```
[root@serverX ~]# chage -M 90 -I 30 -E 2011-09-30 hugo
```



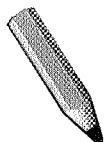
Practice Quiz

Collaborative Directory Permissions

1. What command would change the permissions on a directory to be a private single group collaborative directory?
chmod 2770 /directory

2. What command would grant a second group access to that directory?
setfacl -m g:group:rwx /directory

3. What command would grant that second group read-write access to any newly created files in that directory?
setfacl -m d:g:group:rw /directory



Test

Criterion Test

Exercise

Using ACLs to Grant and Limit Access

Using the users and groups created earlier on serverX....

If you do not already have the users and groups defined, run **lab-add-users** on serverX.

Graduate Students need a directory /opt/research, where they can store generated research results. Newly created files in the directory should have the following properties:

1. The files should be group owned by the group grads.
2. Professors (members of the group profs) should have read/write access to the directory.
3. Summer interns (members of the group interns) should have read-only access to the directory.

4. Additionally, other users (not a member of profs, grads, or interns) should not be able to access the directory at all.

```
mkdir /opt/research
chgrp grads /opt/research/
chmod g=rwx /opt/research/
setfacl -m g:profs:rwx /opt/research/
setfacl -m g:interns:rx /opt/research/
setfacl -m d:g:profs:rwx /opt/research/
setfacl -m d:g:interns:rx /opt/research/
setfacl -m d:g:grads:rwx /opt/research/
```

Authentication Management



Practice Quiz

LDAP Client Configuration

1. What seven pieces of information are typically provided by *User account information* services?
username:password:UID:GID:GECOS:/home/dir:shell

2. What "other" type of information can be provided by a *network directory service*?
Authentication method

3. What are the three pieces of information a client machine needs to be configured to get user information from an LDAP directory service?
Server's fully-qualified hostname, Base DN, and CA certificate

4. What does the command **getent passwd ldapuser1** do? Why is this useful?
The getent command looks up the user information for ldapuser1 from the password file database. This confirms that a system is properly configured as an LDAP client.



Practice Performance Checklist

Kerberos Configuration Exercise

You will modify your previous LDAP-based configuration to now use only Kerberos for authentication. LDAP will still be used to provide account information.

- ☐ Log into serverX and escalate privileges to root

```
[student@serverX ~]$ su -  
Password:  
[root@serverX ~]#
```

- ☐ Verify necessary packages are installed

```
[root@serverX ~]# rpm -q krb5-workstation  
krb5-workstation-1.8.2-3.el6.x86_64
```

If **krb5-workstation** is not installed, use **yum** to install it:

```
[root@serverX ~]# yum install -y krb5-workstation
```

- ❑ Configure system to use the following LDAP and Kerberos settings:
 - LDAP Server: instructor.example.com (uses TLS)
 - LDAP Certificate: `ftp://instructor.example.com/pub/EXAMPLE-CA-CERT`
 - LDAP Base DN: `dc=example,dc=com`
 - Kerberos Realm: `EXAMPLE.COM`
 - Kerberos KDC: `instructor.example.com`
 - Kerberos Admin Server: `instructor.example.com`
 - Be sure the **sssd** service is enabled

Launch **system-config-authentication**. Select LDAP for the User Account Database and Kerberos password for Authentication Method. Provide the information above for each of the two services. Once you complete the information and apply your changes, the sssd service should be started. To confirm the service is running, do the following:

```
[root@serverX ~]# service sssd status
sssd (pid 2634) is running...
```

- ❑ Test the change by logging in to serverX with ssh:
 - Username: **ldapuserX** (where X is your station number)
 - Password: **kerberos**

The testing can be performed via ssh or by switching to an alternate virtual console on virt-manager. Note that the user's home directory will not be available until the automounter is configured later in this unit.



Practice Quiz

Troubleshooting Authentication Quiz

1. How does one normally configure SSSD? Authentication Configuration Tool, **authconfig**, or edit **/etc/sss/sss.conf**
2. Which directory holds log messages from sssd? **/var/log/sss/**
3. How can we increase the logging detail that is generated? Change **/etc/sss/sss.conf** and set **debug_level=[0-10]** where higher means more detail. This is set in individual "service" areas in the file, allowing for unique levels.

4. When you cannot log in to correct an authentication misconfiguration, what approaches are available to you? Single-user mode or runlevel 1



Practice Performance Checklist

Use a NFS home directory server to provide automounted home directories.

The university also provides a NFS home directory server for its undergraduates. Use the NFS home directory server to automount the home directories of the previously defined users.

Here is information about the home directory server.

- Hostname: *instructor.example.com*
- Exported Directory: **/home/guests/**
- ☐ Extend the configuration of your automounter to mount to the **/home/guests** directory.

Add the following line to **/etc/auto.master**:

```
/home/guests /etc/auto.guests
```

- ☐ Have the automounter attempt to map any specified target directory as the analogous directory from the home directory server.

As an example, a request to access the local directory **/home/guests/ldapuser1** should attempt to mount the directory **/home/guests/ldapuser1** from *instructor.example.com*.

Create a file called **/etc/auto.guests** with the following content:

```
* instructor.example.com:/home/guests/&
```

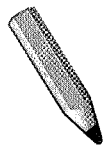
- ☐ Have the automounter service reload its configuration files.

```
[root@serverX ~]# service autofs reload
```

- ☐ From another terminal, attempt to shell into your remote server as the user **ldapuserX** with a password of **password**. The user's home directory should be automatically mounted.

- ☐ Run the **lab-grade-autofshomes** script when you are complete to verify your work.

```
[root@serverX ~]# lab-grade-autofshomes
```



Test

Criterion Test

Case Study

Enhance User Security

Before you begin...

Make sure to run the **lab-setup-taylorlocke** from your desktopX system, which will prepare your serverX system for the lab.

Taylor and Locke, a prestigious law firm, recently hired a security consultant to advise them regarding their servers. As the law firm's servers hold sensitive client information, security is a priority!

The security consultant recommended that all servers use LDAP for centralized accounts and Kerberos for authentication. Overall, the LDAP/Kerberos deployment went well. However, one of the servers that you manage appears to be mis-configured.

Correct the configuration on serverX so that LDAP users are able to login with Kerberos authentication (details below).

- LDAP Server: instructor.example.com (uses TLS)
- LDAP Certificate: ftp://instructor.example.com/pub/EXAMPLE-CA-CERT
- LDAP Base DN: dc=example,dc=com
- Kerberos Realm: EXAMPLE.COM
- Kerberos KDC: instructor.example.com
- Kerberos Admin Server: instructor.example.com

Test the change by logging in to serverX with ssh:

- Username: ldapuserX (where X is your station number)
- Password: kerberos

Once LDAP users can login, configure autofs to provide automounted home directories. The home directories are shared from instructor.example.com.

When you are ready, run the **lab-grade-taylorlocke** script on serverX to check your work.

1. Take serverX to single user mode.
2. Configure LDAP/Kerberos authentication:

```
# authconfig --enableldap --ldapserver=instructor.example.com --enableldaptls
--ldaploadcacert=ftp://instructor.example.com/pub/EXAMPLE-CA-CERT
--ldapbasedn="dc=example,dc=com" --disableldapauth --enablekrb5
--krb5kdc=instructor.example.com --krb5adminserver=instructor.example.com
--krb5realm=EXAMPLE.COM --enablesssd --enablesssdauth --update
```

3. Take serverX to multi-user mode, either runlevel 3 or 5.
4. Add the following line to **/etc/auto.master**:

```
/home/guests    /etc/auto.home
```

Create and add the following to **/etc/auto.home**:

```
*    -rw,hard,intr    instructor.example.com:/home/guests/&
```

Make the automounter reload its configuration:

```
[root@serverX ~]# service autofs reload
```

Installation, Kickstart and Virtualization



Practice Performance Checklist

Modify a Kickstart File without system-config-kickstart

Once you complete this exercise, you will have taken all of the steps necessary to Kickstart a new system, short of the actual installation. You will perform a Kickstart installation later in this unit. Perform the following steps on desktopX:

- Create a copy of `/root/anaconda-ks.cfg` called `~/projman.cfg`. Using only a text editor, modify that file so it meets the following criteria. The installation must be fully automated, and exactly like the basic workstation installation, except...

- Perform the following disk partitioning:
 - Initialize the MBR if necessary
 - Clear all existing partitions
 - `/boot` (ext4) - 200 MB
 - swap - 512 MB
 - `/` (ext4) - all remaining space (5 GB minimum)

The partitioning directives would look like the following in the starting section of the Kickstart file:

```
# Partitioning according to lab specifications:
zerombr
clearpart --all
part /boot --fstype=ext4 --size=200
part swap --size=512
part / --fstype=ext4 --size=5000 --grow
```

- The **E-mail server** package group should be installed

The `%packages` section should contain the following line:

```
@E-mail server
```

- The **fetchmail** package, which is not included with the **E-mail server** group by default, should be installed

The `%packages` section should also contain the following line:

```
fetchmail
```

- Be sure to remove existing scripting from `%pre` and `%post`

- Use **echo** to append the following text to the end of **/etc/issue**:

```
PROJECT MANAGEMENT
```

The **%post** section should look like the following:

```
%post
echo 'PROJECT MANAGEMENT' >> /etc/issue
%end
```

- ☐ **ksvalidator** must be able to validate the file

No output should display when running the following command:

```
[root@desktopX ~]# ksvalidator projman.cfg
```

- ☐ When complete, publish the file so it can be used for an installation. Deploy a web server on desktopX and copy **projman.cfg** to **/var/www/html/**.

Take the steps necessary to deploy a web server. Install the **httpd** package if necessary then start the service and configure it to start at boot time:

```
[root@desktopX ~]# yum install -y httpd
[root@desktopX ~]# service httpd start
[root@desktopX ~]# chkconfig httpd on
```

Make sure the Kickstart file is readable then publish it:

```
[root@desktopX ~]# chmod 644 projman.cfg
[root@desktopX ~]# cp projman.cfg /var/www/html/
```

- ☐ Use a web browser to confirm your Kickstart file is readable. The URL you use to view the file is what you would pass to the installer with the **ks=URL** argument.

Launch firefox and point it to **http://desktopX.example.com/projman.cfg**. The Kickstart file should display if you have published it correctly.

Facts about KVM Virtualization

- KVM = Kernel-based Virtual Machine
- KVM requirements:
- VirtIO support = paravirtualized drivers used by KVM guests (provide better IO performance)
- KVM benefits include:
- Implemented as a kernel module
- Allows a standard Linux kernel to act as a hypervisor

- 64-bit AMD or Intel processors
- Hardware assisted virtualization extensions
- 64-bit operating system
- High performance
- Simple design
- Adopted by upstream kernel developers
- **grep flags /proc/cpuinfo**
- Relevant flags include:
 - **lm** = Long Mode (64-bit x86)
 - **svm** = Secure Virtual Machine (AMD)
 - **vmx** = Virtual Machine Extensions (Intel)



Practice Quiz

Introduction to KVM Virtualization

1. Hardware-assisted virtualization requires a special CPU with virtualization enabled in the BIOS.
(select one of the following...)
 - a. True
 - b. False
2. KVM is a kernel-based virtualization technology that allows both Linux and Windows to be installed as a virtual machine without using a special kernel.
(select one of the following...)
 - a. True
 - b. False
3. KVM is becoming popular because it achieves high performance because of its complex design.
(select one of the following...)
 - a. True
 - b. False

KVM is actually simple in its design.
4. The **lm** and either **svm** or **vmx** CPU flags are required for kernel-based virtualization.
(select one of the following...)
 - a. True

- b. False
5. KVM will work on both 32-bit and 64-bit hardware.
(select one of the following...)
a. True
b. False
- KVM must run on a host with a 64-bit operating system running on 64-bit hardware (that is what the `lm CPU` flag represents) Guest virtual machines may be x86 32-bit.*
6. Upstream software developers have adopted KVM into the source code for the Linux kernel.
(select one of the following...)
a. True
b. False



Practice Performance Checklist

Virtual Guest Installation

In this lab you will install a new virtual machine with Red Hat Enterprise Linux using **virt-manager** and the graphical installer. Once you have successfully completed the lab you will need to remove both the virtual machine and its logical volume to reclaim system resources needed for other labs.

Perform the following steps on desktopX:

- ☐ Gracefully shutdown your serverX virtual machine to reclaim system CPU and RAM resources.

Launch **virt-manager** by selecting **Applications** → **System Tools** → **Virtual Machine Manager**. Right-click on the icon for the **vserver** virtual machine then select **Shut Down** → **Shut Down**.
- ☐ Create a logical volume 10 GB in size from the **vol10** volume group and name it **guest**.


```
[root@desktopX ~]# lvcreate -n guest -L 10G vol10
```
- ☐ Create a Red Hat Enterprise Linux 6 virtual machine with the following characteristics:
 - Name = guest
 - Install media = network install from <http://instructor.example.com/pub/rhel6/dvd>
 - Memory (RAM) = 768 MB
 - CPUs = 1
 - Storage device = the logical volume created in the previous step

- Network - use DHCP to get IP address

Within **virt-manager**, right-click on the **localhost (QEMU)** item and select **New**. When the "New VM" dialog box appears, type **guest** for the name and choose the **Network Install (HTTP, FTP, or NFS)** radio button for installation method. Click the **Forward** button when you are ready to proceed.

Type **http://instructor.example.com/pub/rhel6/dvd** in the URL field. Click the **Forward** button when you are ready to proceed. If a warning dialog box appears cautioning about the permissions of **/home/student/.virtinst/boot**, then click **Yes** and move on.

In the next dialog box, select **768 MB** for **Memory (RAM)** and leave the **CPUs** set to 1. Click the **Forward** button when you are ready to proceed.

For storage, select the **Select managed or other existing storage** radio button, then specify the **/dev/vol0/guest** pathname. Click the **Forward** button when you are ready to continue.

After reviewing the final dialog box, click **Finish** to complete the creation of the virtual machine and begin your interaction with the Red Hat installer, Anaconda.

When the text-based menus appear, select the appropriate language and keyboard choices for your locale. Each time choose **OK** to proceed to the next menu. Once the network settings have been specified, the graphical installer will appear. Select **View → Resize to VM** from the **virt-manager** menus.

- Once **Anaconda** launches, build your guest system according to the following specifications:
 - Use the entire virtual drive with a default disk partitioning scheme
 - Assign **redhat** as the root password
 - Install the Desktop package group

Click the **Next** button to move beyond the introductory screen.

On the storage screen, make sure the **Basic Storage Devices** radio button is selected and click **Next**. If a **Warning** dialog box appears suggesting the storage needs to be reinitialized, click the **Re-initialize all** button to wipe the virtual machine's drive.

When the network configuration screen appears, leave the default hostname chosen. The network will be configured because a network installation is being performed. Click then **Next** button to continue.

Choose an appropriate timezone and make sure the **System clock uses UTC** checkbox is checked. Click **Next** to continue.

Specify the root password of **redhat** twice then click the **Next** button. When the **Weak Password** dialog box appears, ignore the warning and click the **Use Anyway** button to continue.

Since the problem exercise said to use the default partitioning scheme, click the **Next** button to advance past the disk partitioning screen. Click the **Write changes to disk** button when the warning dialog box appears. You will see the disk get partitioned and formatted at this point.

The software selection screen will appear next. Select the radio button for the **Desktop** package group instead of the default **Basic Server**. Click the **Next** button to continue. After the software dependency checks complete the installation will begin.

- ☐ Reclaim the system resources used by this lab exercise. Remove the virtual machine you created and the storage it uses.

Use **virt-manager** to force the virtual guest to shutdown. Right click on the virtual machine and select **Delete** to remove the system profile. Finally execute the following command to reclaim the disk resources use by the virtual guest:

```
[root@desktopX ~]# lvremove /dev/vol0/guest
Do you really want to remove active logical volume guest? [y/n]: y[enter]
```

Commands Used to Manage Virtual Machines

Until now **virt-manager** has been used to manage virtual machines. There is a command-line tool, **virsh**, that implements the same functionality as **virt-manager** without the need for a GUI. Both of these utilities use the **libvirt** library so they can be used interchangeably to manage virtual machines.

1. Power on a virtual machine: **virsh start name**
2. Gracefully shut down a virtual machine: **virsh shutdown name**
3. Power off a virtual machine: **virsh destroy name**
4. Connect to a console of a virtual machine: **virsh console name**
5. Disconnect from a console of a virtual machine: **Ctrl+]**
6. Start a virtual machine at boot time: **virsh autostart name**



Practice Performance Checklist

Virtualization Commands

Perform all of the following tasks from the command line on desktopX. Do not use **virt-manager** or **virt-viewer** during this lab exercise.

- ☐ Use **virsh list --all** to determine the virtual domain ID (or name) of serverX. You will need the domain name to perform the following steps.

```
[root@desktopX ~]# virsh list --all
 Id Name                               State
-----
```

- vserver shut off

- ☐ If serverX is not running, power it on.

```
[root@desktopX ~]# virsh start vserver
Domain vserver started
```

- ☐ Gracefully shut down serverX.

```
[root@desktopX ~]# virsh shutdown vserver
Domain vserver is being shutdown
```

- ☐ Power on serverX.

```
[root@desktopX ~]# virsh start vserver
Domain vserver started
```

- ☐ Connect to the console of serverX.

```
[root@desktopX ~]# virsh console vserver
Connected to domain vserver
Escape character is ^]
```

- ☐ The virtual machine may not be configured to present a console on the virtual console. Disconnect from the console.

Type **Ctrl+]** to exit the virtual console.

- ☐ Power off serverX.

```
[root@desktopX ~]# virsh destroy vserver
Domain vserver destroyed
```

- ☐ Ensure serverX starts at boot time.

```
[root@desktopX ~]# virsh autostart vserver
Domain vserver marked as autostarted
```



Test

Criterion Test

Performance Checklist

Kickstart a Virtual Machine

- ☐ Copy the **/root/anaconda-ks.cfg** file from serverX to desktopX and call it **~/test.cfg**. Shutdown serverX after you have copied the file to reclaim system resources for the rest of the lab.

- ❑ Modify **test.cfg** according to the following criteria:
 - Partition storage according to the following:
 - /boot (ext4) 200 MB
 - swap 512 MB
 - / (ext4) 8 GB
 - Add the **gimp** package
 - Create a **/root/install-date** file with the date and time.

Add the following to the Kickstart file

```
clearpart --all
part /boot --fstype=ext4 --size=200
part swap --size=512
part / --fstype=ext4 --size=8192
....
[after %packages]
gimp
...

%post
date > /root/install-date
%end
```

- ❑ Copy **test.cfg** to **/var/www/html/** on desktopX. Make sure the file is readable by Apache. Start the **httpd** daemon if it is not already running.

```
[root@desktopX ~]# cp ~/test.cfg /var/www/html/
[root@desktopX ~]# chmod 644 /var/www/html/test.cfg
[root@desktopX ~]# service httpd restart
```

- ❑ Create a logical volume in the volume group **vol0** named **test** large enough to serve as the disk for your virtual machine .

```
[root@desktopX ~]# lvcreate -n test -L 10G vol0
```

- ❑ Start a virtual machine installation using your **test.cfg** Kickstart file. Name the virtual machine **test**. Use the install media from <http://instructor/pub/rhel6/dvd> and allocate the virtual machine to have 768 MB of RAM and 1 CPU. Use the logical volume you created in the previous step as the storage for your virtual machine.
- ❑ Reboot your virtual machine when it is finished installing and confirm that it installed correctly.
- ❑ **IMPORTANT:** Delete your virtual machine and the logical volume it uses for storage to reclaim resources needed in future labs.

```
[root@desktopX ~]# virsh destroy test
[root@desktopX ~]# virsh undefine test
```

```
[root@desktopX ~]# lvremove -f /dev/vol0/test
```

Boot Management



Practice Performance Checklist

Resolve GRUB issues

- ☐ Run the **lab-setup-bootbreak** script on desktopX to prepare your virtual server for boot time problems.

```
[root@desktopX ~]# lab-setup-bootbreak
```

- ☐ After serverX has booted, run the **lab-setup-bootbreak-5** script on serverX to introduce an issue with its boot process.

```
[root@serverX ~]# lab-setup-bootbreak-5
```

- ☐ Reboot serverX and modify the bootloader temporarily so the system can boot and you can log in.

1. Interrupt the GRUB countdown: **Esc** key
2. Use "e" to edit current configuration
3. Select **initrd** line to correct with arrow keys
4. Type "e" again to edit the current line, removing the "**-BROKEN**" phrase
5. Type "b" to boot with the current changes
6. Confirm that you can again log in



Practice Performance Checklist

Resolve GRUB issues persistently

- ☐ Reboot and ensure the issue from the previous problem is persistent. You will need to apply the fix as before to boot the system.

Interrupt the GRUB countdown (Esc key). Use "e" to edit current configuration. Select **initrd** line to correct with arrow keys. Type "e" again to edit the current line, removing the "**-BROKEN**" phrase. Type "b" to boot with the current changes.

- ☐ Edit the configuration file to fix the issue permanently.

Edit **/boot/grub/grub.conf** and permanently remove "**-BROKEN**" from the **initrd** line.

- ☐ Install a new kernel from the **Errata** repository.

```
[root@serverX ~]# yum update kernel
```


- ☐ Revert to the older kernel. In other words, with the new kernel still available, ensure that when you reboot, the older kernel is the default kernel.

Edit **/boot/grub/grub.conf** and set **default=1** so that the old kernel will be booted by default.

- ☐ Reboot the system to confirm that the old kernel successfully boots and you are able to log in.

```
[root@serverX ~]# reboot
```



Practice Performance Checklist

Change the default runlevel

You are configuring a new system that you will be accessing remotely. The system is currently booting into run level 5 by default, but this machine will be housed in a data center where you will only login to it remotely. You want to change the serverX system to boot to runlevel 3 by default.

- ☐ Configure the system to boot into runlevel 3 by default.

Change the line in **/etc/inittab** to the following:

```
id:3:initdefault:
```

- ☐ Reboot, then verify the current runlevel.

```
[root@serverX ~]# reboot
```

```
...
```

```
[root@serverX ~]# who -r
```

```
run-level 3 2010-12-13 10:26
```



Practice Performance Checklist

Changing the root Password

This timed drill is designed to give you practice changing the root password on a system with an unknown root password.

- ☐ Begin by running the **lab-setup-bootbreak-4** script on serverX. This will change the root password to something unknown and mark the current time.

```
[root@serverX ~]# lab-setup-bootbreak-4
```

- ☐ Get into the system and reset the root password to **redhat**.



Note

At the release of Red Hat Enterprise Linux 6, there was an SELinux bug which blocked the **passwd** command in single-user mode (#644820). If you have the original *selinux-policy* package installed, you must run the **setenforce 0** command in runlevel 1 before the **passwd** command for it to work. After changing the password you should run **setenforce 1** again to put SELinux back in enforcing mode.

Interrupt the GRUB countdown (Esc key). Use "e" to edit current configuration. Select **kernel1** line to correct with arrow keys. Type "e" again to edit the current line, appending a "space" and "**single**". Type "b" to boot with the current changes.

```
# setenforce 0
# passwd
Changing password for user root.
New password: redhat
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password: redhat
passwd: all authentication tokens updated successfully.
# setenforce 1
```

- ☐ Once you have reset the password, change the system into runlevel 5 and run the **lab-grade-bootbreak-4** script on serverX.

```
# init 5
...
[root@serverX ~]# lab-grade-bootbreak-4
```

- ☐ View the feedback from the script to ensure you completed the task correctly. The grading script will display a time, write it down.
- ☐ Repeat the process again at least five times.
- ☐ Circle your best time.



Practice Exercise

Using the Rescue Environment

Before you begin...

Run **lab-setup-bootbreak** on desktopX.

This timed drill is designed to give you practice accessing the rescue environment. The installation path is <http://instructor.example.com/pub/rhel6/dvd>. The path for individual packages is <http://instructor.example.com/pub/rhel6/dvd/Packages/>

1. After serverX has booted, run the **lab-setup-bootbreak-0** script. This script will alter your system and cause difficulties booting.

Try booting the system without the kernel arguments of **rhgb quiet**. Note the following error messages (Shift-PgUp can be used to scroll the screen back up):

```
/etc/rc.d/rc.sysinit/: line 26: mount: command not found
readahead: starting
        Welcome to Red Hat Enterprise Linux Server
...
Remounting root filesystem in read-write mode: /etc/init.d/functions: line 536:
mount: command not found
...
```

2. Boot into the rescue environment to diagnose and resolve the issue.

- Boot into rescue mode.

Rather than booting our virtual machine from an Installation DVD, cause it to boot from the network by pressing <Ctrl-B> at the gPXE dialog.

```
gPXE> autoboot
```

Choose "Rescue installed system"

Choose Language and Keyboard Type as appropriate

Choose "URL" for Rescue Method

Use DHCP for IPv4 only

<http://instructor.example.com/pub/rhel6/dvd> for URL Setup

"Continue" to mount your hard drive under **/mnt/sysimage**

Finally, open a "shell"

- Verify problem based on earlier error message:

```
bash-4.1# chroot /mnt/sysimage
sh-4.1# mount
sh: mount: command not found
sh-4.1# yum provides /bin/mount
... output omitted ...
sh-4.1# yum reinstall util-linux-ng
... output omitted ...
sh-4.1# mount
... output omitted ...
```

3. Confirm the problem has been solved by rebooting the system.

```
sh-4.1# exit
exit
bash-4.1# exit
```

Choose "reboot".

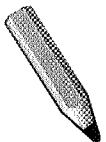
4. Repeat this process as often as possible during the allotted time.



Practice Quiz

Troubleshooting Quiz

1. In maintenance mode, run `mount -o remount,rw /` to mark the / partition as writable.
2. Assuming you have only one hard drive, and the first partition contains /boot, if you have minor MBR corruption, fix the corruption issue by booting into rescue mode, then run the grub command. Type root (hd0,0) followed by setup (hd0), then exit.
3. If you have filesystem corruption issues, the machine will boot into maintenance mode.
4. In maintenance mode, run fsck to fix filesystem corruption issues.



Test

Criterion Test

Exercise

Troubleshooting the Boot Process

Before you begin...

Run the **lab-setup-bootbreak** command on desktopX to setup the lab.

This practice exercise includes three break/fix challenges. For each, your serverX virtual machine will be modified in some way that prevents it from booting correctly and you will diagnose and correct the problem.

1. Run **lab-setup-bootbreak-1** on serverX. After running the script, serverX should no longer boot correctly. Diagnose and correct the problem. You will know you have the solution when serverX boots normally again.

lab-setup-bootbreak-1 replaces **UUID** with **UID** in **/etc/fstab**. Notice that **vim** will highlight that error in red. To fix the issue, make the / filesystem writable, then change **UID** to **UUID** in **/etc/fstab**.

-
2. When you have resolved the first scenario, repeat the process with **lab-setup-bootbreak-2** and **lab-setup-bootbreak-3**.

lab-setup-bootbreak-2 changes the default runlevel to 9. To fix this issue, boot into a standard runlevel (by adding **3** to the kernel line, for example), then edit **/etc/inittab** and change **id:9:initdefault:** to **id:3:initdefault:**

lab-setup-bootbreak-3 introduces a misspelling in **/boot/grub/grub.conf**. To fix this issue, edit the kernel line from the grub prompt and change **rot=** to **root=**. Once the system boots, make the same change in **/boot/grub/grub.conf**.

SELinux Management



Practice Quiz

Basic SELinux Concepts

1. To which of the following does SELinux apply security context (check all that apply)?

(select one or more of the following...)

- a. Ports
- b. Processes
- c. Files
- d. Directories
- e. Remote file systems

2. SELinux can be used to:

(select one or more of the following...)

- a. Protect a service from running on other ports.
- b. Protect user data from applications like the web server
- c. Block remote systems from accessing local ports

This describes a firewall.

- d. Keep the system updated

This describes something like Red Hat Network.

- e. Access a web server

This describes a web browser like Firefox.

3. Which of the following are standard SELinux context types?

(select one or more of the following...)

- a. selinux_type

This is non-existent.

- b. object_r

This is an SELinux role.

- c. httpd_sys_content_t
- d. tmp_t
- e. user_u

This is an SELinux context user.



Practice Quiz

SELinux Modes

1. SELinux permissive mode allows logging, but not protection.

2. SELinux enforcing mode protects the system.
3. Which of the following are valid SELinux modes?

(select one or more of the following...)

- a. enforcing
- b. testing
- c. permissive
- d. disabled
- e. logging



Practice Exercise

Correcting SELinux file contexts

You have been asked to adjust your remote machine's DNS configuration to exactly match the configuration from your desktop machine. You decide the easiest way is to copy the file **/etc/resolv.conf** from the local machine to the remote machine.

1. Transfer the **/etc/resolv.conf** file from your desktop machine to *root*'s home directory on serverX.

```
scp /etc/resolv.conf root@serverX:
```

2. Shell into serverX as **root**. All of the following steps should occur on your server.
3. Observe the SELinux context of the initial **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

Original **/etc/resolv.conf** context: system_u:object_r:net_conf_t:s0

4. Move **resolv.conf** from *root*'s home directory to **/etc/resolv.conf**.

```
mv /root/resolv.conf /etc
```

5. Observe the SELinux context of the newly copied **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

New **/etc/resolv.conf** context: unconfined_u:object_r:admin_home_t:s0

6. Restore the SELinux context of newly positioned **/etc/resolv.conf**.

```
restorecon /etc/resolv.conf
```

7. Observe the SELinux context of the restored **/etc/resolv.conf**.

```
ls -Z /etc/resolv.conf
```

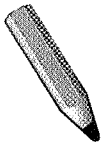
Restored **/etc/resolv.conf** context: system_u:object_r:net_conf_t:s0



Practice Quiz

Monitoring SELinux Violations

1. What file contains log entries providing unique identifiers for SELinux violations? **/var/log/audit/audit.log**
2. Given the UUID of an SELinux violation, what command generates a text report of the problem? **sealert -l UUID**



Test

Criterion Test

Exercise

Managing SELinux

Before you begin...

Before you begin, run the **lab-setup-selinux** command on desktopX

1. Login to serverX as **student**. Open a terminal and switch to the **root** user.
2. Copy the *web_content.tgz* archive from *instructor:/var/ftp/pub/materials* to */tmp*.

```
[root@serverX ~]# cp /net/instructor/var/ftp/pub/materials/web_content.tgz /tmp
```

3. Extract the archive into */tmp*.

```
[root@serverX ~]# cd /tmp
[root@serverX tmp]# tar -xvf web_content.tgz
```

4. Move the extracted directory to */var/www/html*.

```
[root@serverX tmp]# mv web_content /var/www/html/
[root@serverX tmp]# cd
```

5. Start the web service.

```
[root@serverX ~]# service httpd start
```

6. Try to observe the new directory with your web browser by visiting the URL *http://serverX/web_content*.

```
[root@serverX ~]# elinks -dump http://serverX/web_content
```

7. Search your system for the UUIDs of any SELinux violations your attempt to browse the newly installed content might have generated.


```
[root@serverX ~]# cat /var/log/messages | grep 'sealert -l'
```

8. Generate text reports for the violations.

```
[root@serverX ~]# sealert -l UUID > ~/httpd_selinux.log
```

Where *UUID* is the UUID given in `/var/log/messages`

9. Follow the report's advice to restore the SELinux contexts of the newly installed content.

Find the **Fix Command** section in `~/httpd_selinux.log`

```
[root@serverX ~]# restorecon -Rv /var/www/html/web_content/
```

10. Confirm that you can view the material from your web browser by visiting the URL `http://serverX/web_content`.

```
[root@serverX ~]# elinks -dump http://serverX/web_content
```

Firewall Management

- Rule - criteria determining which packets to match and a target, or action, determining what to do with those packets.
- Chain - a list of *rules* which will be checked in order, first match takes effect.
- Policy - the default action, **ACCEPT** or **DROP**, taken if no *rule* matches in a built-in *chain*.
- Table - a set of *chains* used for a particular purpose: **filter** to block traffic, **nat** to modify the destination or apparent source of a packet.
- INPUT - packets addressed to the firewall
- OUTPUT - packets originating from a service on the firewall (not forwarded)
- FORWARD - packets originating from another machine, that are not addressed to the firewall but are being forwarded (routed) elsewhere (when **net.ipv4.ip_forward=1**)
- ACCEPT - the packet passes the chain
- DROP - the packet is dropped as if it was never seen
- REJECT - the packet is rejected, and the firewall sends an error message (an ICMP port unreachable message by default)
- LOG - information about the packet is logged to syslog; we go on to the next rule in the chain



Practice Exercise

Implement a Firewall

In this exercise you will implement a firewall on serverX that rejects all packets, except that it will allow ICMP traffic for example.com and allow SSH for everyone except remote.test.

1. Log into serverX as **root** using **virt-viewer** or **virt-manager**.
2. Create a simple deny all (except loop back) firewall by creating **/root/bin/resetfw.sh** that
 1. sets the **INPUT** chain's default policy to **DROP**,
 2. flushes all rules in the filter table, and
 3. will **ACCEPT** all packets from the loopback interface

```
[root@serverX ~]# cat /root/bin/resetfw.sh
#!/bin/bash
# Set INPUT chain default policy to DROP
iptables -P INPUT DROP
# Flushes all rule in the filter table
iptables -F
# Will ACCEPT all packets from loopback interface
iptables -A INPUT -i lo -j ACCEPT
```

3. Run your script and record the results of the following:

- **ping** and **ssh** serverX from desktopX and from remoteX.remote.test

Both should fail since only traffic from serverX's loopback interface is being ACCEPTed.

4. What happens when you **ping** desktopX and 192.168.0.X from serverX now? Why?

Again, both should fail since the replies from desktopX and 192.168.0.X are being DROPPed.

5. Enable stateful firewalling by appending to your script a rule that will

- **ACCEPT** all **ESTABLISHED,RELATED** packets

```
[root@serverX ~]# tail -n 2 /root/bin/resetfw.sh
# ACCEPT all ESTABLISHED, RELATED packets
iptables -A INPUT -m state --state ESTABLISHED,RELATED
```

6. Run your script and record the results of the following:

- **ping** desktopX and 192.168.0.X from serverX

These should now work since the "replies" are part of an ESTABLISHED connection. Note, that the reverse is still not possible because the connection never gets started.

7. Reject all packets from remote.test by appending to your script a rule that will

- **REJECT** all packets from the 192.168.1.0/24 network

```
[root@serverX ~]# tail -n 2 /root/bin/resetfw.sh
# REJECT all packets from 192.168.1.0/24 network
iptables -A INPUT -s 192.168.1.0/24 -j REJECT
```

8. Run your script and record the results of the following:

- **ping** and **ssh** serverX from desktopX and from remoteX.remote.test

These should still not work from desktopX (being DROPPed) since there are no applicable rules for the initial inbound packet. The attempts from remoteX.remote.test are being explicitly REJECTed and so will also not work.

9. Enable ICMP traffic for example.com by appending to your script a rule that will

- **ACCEPT** all **icmp** traffic from 192.168.0.0/24

```
[root@serverX ~]# tail -n 2 /root/bin/resetfw.sh
# ACCEPT all icmp traffic from 192.168.0.0/24
iptables -A INPUT -p icmp -s 192.168.0.0/24 -j ACCEPT
```

10. Run your script and record the results of the following:

- **ping** and **ssh** serverX from desktopX

The **ping** should now work since the inbound **icmp** protocol is now ACCEPTed. However, **ssh** still does not work.

11. Enable SSH traffic for all hosts by modifying your script to

- **ACCEPT** all **NEW** connections to **tcp** port **22**

Notice that the direction did NOT say to append the new rule. For this to ACCEPT all NEW connections, we need to insert this into the script *above* the REJECT of 192.168.1.0/24.

```
[root@serverX ~]# cat /root/bin/resetfw.sh
#!/bin/bash
# Set INPUT chain default policy to DROP
iptables -P INPUT DROP
# Flushes all rule in the filter table
iptables -F
# Will ACCEPT all packets from loopback interface
iptables -A INPUT -i lo -j ACCEPT
# ACCEPT all ESTABLISHED, RELATED packets
iptables -A INPUT -m state --state ESTABLISHED,RELATED
# ACCEPT all NEW connections to tcp port 22
iptables -A INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
# REJECT all packets from 192.168.1.0/24 network
iptables -A INPUT -s 192.168.1.0/24 -j REJECT
# ACCEPT all icmp traffic from 192.168.0.0/24
iptables -A INPUT -p icmp -s 192.168.0.0/24 -j ACCEPT
```

12. Run your script and record the results of the following:

- **ssh** to serverX from desktopX and from remoteX.remote.test

These should now work since the "requests" are a NEW connection on port 22. If only desktopX works and remoteX.remote.test does not work, then check the order of your rules in the previous step.

13. Reject packets by default instead of dropping packets by appending to your script a rule that will

- **REJECT** all other traffic

```
[root@serverX ~]# tail -n 2 /root/bin/resetfw.sh
# REJECT all other traffic
iptables -A INPUT -j REJECT
```

14. Run your script and record the results of the following:

- **ping** and **ssh** serverX from desktopX and from remoteX.remote.test

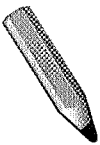
Both **ping** and **ssh** will work from desktopX, but only **ssh** works from remoteX.remote.test.



Practice Quiz

Network Address Translation Quiz

1. The chains available in the **filter** table are **INPUT**, **FORWARD**, and **OUTPUT**
2. The chains available in the **nat** table are **PREROUTING**, **POSTROUTING**, and **OUTPUT**
3. **iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE**
4. **iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.0.1**
5. **iptables -t nat -A PREROUTING -i eth0 -m tcp -p tcp --dport 80 -j DNAT --to-destination 192.168.0.100:8080**
6. The **DNAT** target can only be used in the **PREROUTING** chain and the **OUTPUT** chain of the **nat** table
7. To enable forwarding persistently across reboots add **net.ipv4.ip_forward=1** to **/etc/sysctl.conf** and run **sysctl -p**



Test

Criterion Test

Case Study

The Morris Worm and Fish Supply Company

Before you begin...

Important: Make sure to run the **lab-setup-morrisworm** script on desktopX before you begin! The **lab-setup-morrisworm** script will configure serverX to run on a private network.

The Morris Worm and Fish Supply company is finally looking to modernize its business by opening a website. The web server will run on a private network behind a firewall. The firewall will forward all TCP port 80 traffic to the web server and will perform NAT so that the web server can reach external hosts.

- desktopX.example.com will be the firewall, serverX.example.com will be the web server.
- Configure Apache to run on serverX.example.com. Put some custom content in **/var/www/html/index.html** that will uniquely identify the server.
- Configure the firewall on desktopX to perform NAT that will allow the web server to reach the outside network. You will be able to successfully **ping** instructor.example.com from serverX to confirm this works.

- Finally configure the firewall to forward all TCP port 80 traffic sent to it to the web server running on serverX. You will need to identify serverX's IP address to complete this step. Confirm this works by using a web browser from an external machine, NOT desktopX, to browse <http://desktopX.example.com>.

After you have successfully completed the lab, run **lab-cleanup-morrisworm** on desktopX to reset your network back to its original state.

1. Determine the IP address of serverX

```
[root@serverX ~]# ip a show eth0
```

2. Deploy a web server on serverX

```
[root@serverX ~]# yum install httpd
[root@serverX ~]# service httpd start ; chkconfig httpd on
[root@serverX ~]# echo serverX > /var/www/html/index.html
```

3. On desktopX set **net.ipv4.ip_forward=1**

```
[root@desktopX ~]# sysctl -w net.ipv4.ip_forward=1
```

4. On desktopX add these rules to the firewall (replace 192.168.122.Z with the IP found in the first step.

```
[root@desktopX ~]# NewServerIP=192.168.122.Z
[root@desktopX ~]# iptables -t nat -A POSTROUTING -s ${NewServerIP} -j MASQUERADE
[root@desktopX ~]# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination ${NewServerIP}
```

NTP Server Configuration



Practice Quiz

Configuring NTP Quiz

Answer the questions below based upon the following NTP configuration file:

```
#/etc/ntp.conf

restrict default kod nomodify notrap nopeer noquery
restrict -6 default ignore

restrict 192.168.0.0 mask 255.255.255.0 nomodify notrap nopeer
restrict 192.168.0.101 kod nomodify notrap
restrict 192.168.0.200

server 192.168.0.2
server 192.168.0.3
peer 192.168.0.101
```

1. The NTP client's time is off by 15 minutes, it will eventually sync with the servers.

(select one of the following...)

- a. True
- b. False

2. The NTP client will use the computer's RTC (BIOS) as a time source.

(select one of the following...)

- a. True
- b. False

3. 192.168.0.200 will be able to modify the time on this NTP server.

(select one of the following...)

- a. True
- b. False

4. 192.168.0.4 will be able to query this NTP server.

(select one of the following...)

- a. True
- b. False

5. 192.168.0.3 will be able to use this NTP server as a peer.

(select one of the following...)

- a. True
- b. False

6. Anyone with an IPv4 address will be able use this NTP server as a time source.

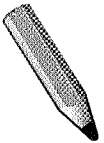
(select one of the following...)

- a. True
- b. False

7. Anyone with an IPv6 address will be able use this NTP server as a time source.

(select one of the following...)

- a. True
- b. False



Test

Criterion Test

Case Study

NTP Server Configuration

Before you begin...

Run **lab-setup-howsonclock** on desktopX.

Howson Heavy Machine and Clock Manufacture, maker of clock tower parts and accessories, recently conducted an audit of all computer systems. The audit revealed several systems with out of sync clocks, including your serverX.example.com machine.

Set up NTP on your serverX to be a client of the NTP service running on instructor.example.com.

In order to have additional time sources, work with a few neighbors so that all of your serverX systems are set up to synchronize as NTP peers.

When you have finished, run **lab-grade-howsonclock** on desktopX to check your work.

1. Identify two or three peer machines. In class, you are encouraged to peer with a neighbor, although these instructions will instead use your 3 assigned machines as peers: desktop1 (192.168.1) and host1 (192.168.0.201) for the server running on server1 (192.168.0.101).

Create the following **/etc/ntp.conf** configuration file, and distribute it to all 3 machines.

```
driftfile /var/lib/ntp/drift

restrict default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict 192.168.0.0 mask 255.255.255.0

server 192.168.0.254
peer 192.168.0.1
peer 192.168.0.101
peer 192.168.0.201
```


On each of the machines, edit the file to omit itself from the peer list. For example, on desktop1, remove the peer line for 192.168.0.1.

2. On each of the machines, enable and start the ntp service.

```
[root@server1 ~]# service ntpd restart
[root@server1 ~]# ssh desktop1 service ntpd restart
[root@server1 ~]# ssh host1 service ntpd restart
[root@server1 ~]# chkconfig ntpd on
[root@server1 ~]# ssh desktop1 chkconfig ntpd on
[root@server1 ~]# ssh host1 chkconfig ntpd on
```

3. For each of the machines, use the **ntpq -p** command to monitor the NTP service's interactions with its peers. For the first 5-10 minutes, you should see output similar to the following.

```
[root@server1 ~]# ntpq -p
remote          refid          st t when poll reach  delay  offset  jitter
=====
*instructor.exam LOCAL(0)      11 u  64  64  17   0.533  -0.096  0.223
desktop1.examp1 .INIT.        16 u  19  64   0   0.000   0.000  0.000
server101.examp .INIT.        16 u  48  64   0   0.000   0.000  0.000
```

Hint: the **watch** command can be useful for monitoring the peering process. In each of three terminals, shell to each of the three machines, and run the command **watch -d ntpq -p**. Use **CTRL-C** to cancel the **watch** command when done.

4. After 5-10 minutes, the NTP services should peer, as reflected by the output of the **ntpq -p** command.

```
[root@server1 ~]# ntpq -p
remote          refid          st t when poll reach  delay  offset  jitter
=====
*instructor.exam LOCAL(0)      11 u  61  64  77   0.327   0.083  0.028
+desktop1.examp1 192.168.0.254  12 u   6  64  77   0.392   0.034  0.011
server101.examp  192.168.0.254  12 u  36  64   2   0.420  -0.057  0.000
```

Note that the "st" column, for "stratum", has converted from considering the peers "stratum 16" (completely untrustworthy) to "stratum 12" (one higher than the best known time source, instructor.example.com).

System Logging Service



Practice Case Study

Usage Reports

Use the tool you investigated to create a simple report that logs information to a file.

Once you have used the tool to generate a report, the instructor will have you share the command you used and explain the output with the rest of the class.

```
[root@serverX] ~]# df -h
[root@serverX] ~]# iostat -dNk 2 10
[root@serverX] ~]# vmstat 2 10
```



Practice Exercise

Remote Logging

1. Configure serverX to accept remote log messages using TCP.

Uncomment the following lines in the MODULES section of **/etc/rsyslog.conf**:

```
$ModLoad imtcp.so
$InputTCPServerRun 514
```

Restart **rsyslog**.

```
[root@serverX ~]# service rsyslog restart
```

2. Configure desktopX to send all **info** priority and higher events to serverX using TCP.

Add the following line to the RULES section of **/etc/rsyslog.conf**:

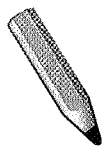
```
*.info @0192.168.0.X+100
```

Restart **rsyslog**.

```
[root@serverX ~]# service rsyslog restart
```

3. Test your configuration.

```
[root@desktopX ~]# logger Test from desktopX
[root@desktopX ~]# tail /var/log/messages
Dec 25 00:00:01 desktopX root: Test from desktopX
[root@serverX ~]# tail /var/log/messages
Dec 25 00:00:01 desktopX root: Test from desktopX
```



Test

Criterion Test

Case Study

System Monitoring and Logs

Before you begin...

Before you begin, run the **lab-setup-blossoms** script on desktopX.

Blossoms, Inc. is a nation wide cooperative of flower and plant growers. Among other things, the cooperative handles IT services for all members. The IT manager has decided to beef up security by requiring remote logging on all servers, including your serverX.

Configure rsyslog on desktopX to accept incoming log messages via UDP from serverX. Then configure **rsyslog** on serverX to send all ***.info** log messages to desktopX via UDP.

When you are ready to check your work, first run **lab-grade-blossoms** on serverX and then run **lab-grade-blossoms** on desktopX.

1. On desktopX, edit **/etc/rsyslog.conf**, removing the comment from lines 13 and 14.

```
# Provides UDP syslog reception
$ModLoad imudp.so
$UDPServerRun 514
```

2. On desktopX, restart the rsyslog service.

```
[root@desktopX ~]# service rsyslog restart
```

3. On desktopX, confirm that **rsyslogd** is bound to the external UDP port 514.

```
[root@desktopX ~]# lsof -i -n -P | grep rsyslogd
rsyslogd 2253    root    3u  IPv4  16673      0t0  UDP *:514
rsyslogd 2253    root    4u  IPv6  16674      0t0  UDP *:514
```

4. On serverX, edit the file **/etc/rsyslog.conf**, inserting the following line, replacing "X" with your station number. Although the exact location doesn't matter, around line 39, near the existing "info" configuration, would be reasonable.

```
*.info                                @desktopX.example.com
```

On serverX, restart the rsyslogd service.

```
[root@serverX ~]# service rsyslog restart
```

5. On desktopX, examine the hostname field (right after the date) in recent log messages to confirm that rsyslog initialization log messages were received from serverX.

```
[root@desktopX ~]# tail -n 4 /var/log/messages
```

```
Dec 9 13:43:39 desktop1 ntpd[1547]: kernel time sync status change 2001
Dec 9 13:50:07 desktop1 ntpd[1547]: synchronized to 192.168.0.254, stratum 3
Dec 9 06:55:25 server1 kernel: imklog 4.6.2, log source = /proc/kmsg started.
Dec 9 06:55:25 server1 rsyslogd: [origin software="rsyslogd" swVersion="4.6.2" x-
pid="24482" x-info="http://www.rsyslog.com"] (re)start
```

Web Service



Practice Performance Checklist

Apache mod_ssl Basics

Deploy an SSL encapsulated Apache web server on serverX. It should use the default self-signed SSL certificate.

- ☐ Log into serverX as root.
- ☐ Install the Apache web server (**httpd**) package, if necessary.

```
[root@serverX ~]# yum install -y httpd
```

- ☐ Install the **mod_ssl** package.

```
[root@serverX ~]# yum install -y mod_ssl
```

- ☐ Examine the **/etc/httpd/conf.d/ssl.conf** configuration file provided by the **mod_ssl** package.

- What is the Apache directive that points to the SSL certificate?
- What is its value?

```
[root@serverX ~]# less /etc/httpd/conf.d/ssl.conf
...
# Point SSLCertificateFile at a PEM encoded certificate.  If
# the certificate is encrypted, then you will be prompted for a
# pass phrase.  Note that a kill -HUP will prompt again.  A new
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file.  Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
...
```

- ☐ Restart the **httpd** service.

```
[root@serverX ~]# service httpd restart
```

- ☐ Launch Firefox and browse to **https://serverX.example.com**. When Firefox presents a warning, take further steps to examine the certificate with Firefox.

- Click the "I Understand the Risks" link.

- Click the "Add Exceptions..." button, then click "View..." when it becomes active.
- Browse the information presented in both the "General" and "Details" tabs.
- Click "Close" when you are finished inspecting the certificate information.



Practice Performance Checklist

Configure Name-Based Virtual Hosts

For this exercise, `wwwX.example.com` is already set up as a **CNAME** alias to `serverX.example.com`.

When you finish the checklist, you will run a grading script, so make sure your web server serves up the content exactly as described in the steps.

- ☐ Create `/var/www/html/index.html` containing the text **"this is serverX."**

```
[root@serverX ~]# echo this is serverX. > /var/www/html/index.html
```

- ☐ From desktopX, use Firefox to verify that the websites `wwwX`, `wwwX.example.com`, `serverX`, and `serverX.example.com` all display your custom **index.html**.

- ☐ Create `/wwwX/html/index.html` containing the text **"this is wwwX."**

```
[root@serverX ~]# mkdir -p /wwwX/html
[root@serverX ~]# echo this is wwwX. > /wwwX/html/index.html
```

- ☐ Modify Apache to enable name-based virtual hosting. `serverX` and `serverX.example.com` should serve `/var/www/html/index.html` as the main page. `wwwX` and `wwwX.example.com` should serve `/wwwX/html/index.html` as the main page.

Add the following content to the bottom of `/etc/httpd/conf/httpd.conf`:

```
# Enable name-based virtual hosting:
NameVirtualHost *:80

# serverX virtual host configuration
<VirtualHost *:80>
    ServerName serverX.example.com
    ServerAlias serverX
    ServerAdmin webmaster@serverX.example.com
    DocumentRoot /var/www/html
    ErrorLog logs/serverX.example.com-error_log
    CustomLog logs/serverX.example.com-access_log common
</VirtualHost>

# wwwX virtual host configuration
<VirtualHost *:80>
    ServerName wwwX.example.com
    ServerAlias wwwX
    ServerAdmin webmaster@wwwX.example.com
    DocumentRoot /wwwX/html
    ErrorLog logs/wwwX.example.com-error_log
    CustomLog logs/wwwX.example.com-access_log common
```

```
</VirtualHost>
```

Have the Apache server reload its configuration:

```
[root@serverX ~]# service httpd reload
```

- ❑ Do not disable SELinux (Hint: You may need to modify the SELinux file context database, or change the SELinux type of certain files).

```
[root@serverX ~]# semanage fcontext -a -t httpd_sys_content_t '/wwwX(/.*)?'
[root@serverX ~]# restorecon -vFR /wwwX
```

- ❑ When you finish, run the **lab-grade-virthost** evaluation script from serverX to make sure you have done everything correctly.

```
[root@serverX ~]# lab-grade-virthost
Good, www1 works
Good, www1.example.com works
Good, server1 works
Good, server1.example.com works
Grading PASSED!
```



Practice Quiz

Apache CGI Quiz

1. CGI stands for
(select one of the following...)
 - a. Content Generated Interface
 - b. Command Gateway Interface
 - c. Common Generated Interface
 - d. Common Gateway Interface

2. The last argument in **ScriptAlias /cgi-bin/ /my/private/cgi-bin/** is
(select one of the following...)
 - a. relative to **DocumentRoot**
 - b. relative to **/var/www/**
 - c. relative to **ServerRoot**
 - d. an absolute path on the filesystem

3. The default **ScriptAlias** in **/etc/httpd/conf/httpd.conf** is pointing to ...
(select one of the following...)
 - a. /var/www/cgi-bin
 - b. /var/html/cgi-bin
 - c. /cgi-bin
 - d. /var/www/html/cgi-bin

4. One of the built in SELinux context types for a generic CGI programs is
(select one of the following...)
- a. `httpd_t`
 - b. `httpd_sys_script_exec_t`
 - c. `script_t`
 - d. `httpd_content_t`
5. The Apache process should have the following filesystem permissions on CGI programs
(select one of the following...)
- a. `---`
 - b. `r--`
 - c. `r-x`
 - d. `rwX`



Practice Performance Checklist

Configure LDAP-Based Authentication

You will configure the web server on serverX with a **/private** URL that is accessible by users in the LDAP directory on instructor.example.com.

- ☐ Configure LDAP authentication on serverX using instructor.example.com as the LDAP server, **dc=example, dc=com** for the base distinguished name and use the certificate found at `ftp://instructor/pub/example-ca.crt`. Choose LDAP passwords.

Run **system-config-authentication** on serverX. Choose **LDAP** in the **User Account Database** drop-down menu. Enter `ldap://instructor.example.com/` as the LDAP Server. Download the certificate from `ftp://instructor/pub/example-ca.crt`. Choose LDAP password as the **Authentication Method**.

- ☐ Login as **root** on serverX. Create a new directory **/var/www/html/private**.

```
[root@serverX ~]# mkdir /var/www/html/private
```

- ☐ In the **private** directory, create an **index.html** containing the text **Private Data**

```
[root@serverX ~]# echo "Private Data" > /var/www/html/private/index.html
```

- ☐ Download **ftp://instructor/pub/example-ca.crt** and place it in **/etc/httpd**

```
[root@serverX ~]# wget ftp://instructor/pub/example-ca.crt -O /etc/httpd/example-ca.crt
```

- ☐ Edit **/etc/httpd/conf/httpd.conf** and add LDAP authentication for the **private** directory.

```
LDAPTrustedGlobalCert CA_BASE64 /etc/httpd/example-ca.crt
```



```
<Directory /var/www/html/private>
    AuthName "Secret Stuff"
    AuthType basic
    AuthBasicProvider ldap
    AuthLDAPUrl "ldap://instructor.example.com/dc=example,dc=com" TLS
    Require valid-user
</Directory>
```

Add the stanza above to the bottom of the `/etc/httpd/conf/httpd.conf` file.

- ☐ Restart Apache

```
[root@serverX ~]# service httpd restart
```

- ☐ Browse to `http://serverX.example.com/private`. You should see an authentication dialog box pop up. If not, close all browser windows, check your configuration, and try again.

```
[root@serverX ~]# elinks http://serverX.example.com/private
```

- ☐ Log in as user `ldapuserX` with a password of `password`

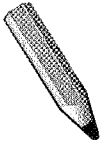
In the dialog box that pops up, enter the username and password above.



Practice Quiz

Troubleshooting Apache Quiz

1. Complete the following command to list all port contexts: `semanage port -l`.
2. Complete the following command to enable Apache to use TCP port 8001: `semanage port -a -t httpd_port_t -p tcp 8001`.
3. The two Apache config file directives to specify the severity (how verbose) error messages are and which file to write to are `LogLevel` and `ErrorLog`.
4. The Apache config file directive to specify the format and location of clients accessing content is `CustomLog`.
5. Full (raw) SELinux AVC messages go to `/var/log/audit/audit.log`.
6. To make SELinux more verbose, you can run `semanage dontaudit off`.
7. The commands to get and set SELinux booleans are `getsebool` and `setsebool`.
8. `man httpd_selinux` will present an SELinux man page specific to Apache.
9. `man -k _selinux` lists all service specific SELinux man pages.
10. The `-F` option to `restorecon` will reset `customizable` types.



Test

Criterion Test 1

Case Study

SSL Encapsulated Web Services

Before you begin...

Run the **lab-setup-hacker** script on desktopX.

Marcelo Hacker is a successful private investigator. In fact, he is doing so well that it is becoming difficult to find the time to meet with prospective clients. Mr. Hacker has decided to set up a website where prospective clients can send him messages. As confidentiality is an important part of the private investigation business, the website must use a signed SSL certificate.

- Set up Apache on serverX to provide an SSL encrypted website for Marcelo Hacker.
- A signed SSL server certificate for your server and a matching key can be found at the following location: **/net/instructor/var/ftp/pub/materials/tls**. Below that directory **certs/serverX.crt** contains the signed certificate for your server and **private/serverX.key** has the private key that matches it.

Deploy the signed certificate for Apache on serverX. Leave the default placeholder website for content. Mr. Hacker will upload his custom content at a later date.

When you have completed the requirements, run **lab-grade-hacker** script on desktopX to check your work.

1. Make sure the **mod_ssl** package is installed.

```
[root@serverX ~]# yum install mod_ssl
```

2. Copy a signed certificate and key from your instructor's "pub" directory, and deploy them into appropriate locations within the **/etc/pki/tls** directory. Confirm that they are readable by the user **apache**, and have an appropriate SELinux context.

```
[root@serverX ~]# cd /net/instructor/var/ftp/pub/materials/tls/
[root@serverX tls]# cp certs/serverX.crt /etc/pki/tls/certs/
[root@serverX tls]# cp private/serverX.key /etc/pki/tls/private/
[root@serverX tls]# cd /etc/pki/tls/
[root@serverX tls]# ls -lZ certs/serverX.crt private/serverX.key
-rw-r--r--. root root unconfined_u:object_r:cert_t:s0 certs/serverX.crt
-rw-r--r--. root root unconfined_u:object_r:cert_t:s0 private/serverX.key
```

3. Confirm that the certificate's Common Name is appropriate for your server.

```
[root@serverX tls]# openssl x509 -text < certs/serverX.crt | grep Subject:
Subject: C=US, ST=North Carolina, O=Example, Inc., CN=serverX.example.com
```

4. Update your server's **/etc/httpd/conf.d/ssl.conf** configuration file, around line 100, setting the **SSLCertificateFile** and **SSLCertificateKeyFile** to refer to your newly install certificate and key file.

```
...
#SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateFile /etc/pki/tls/certs/serverX.crt
...
#SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
SSLCertificateKeyFile /etc/pki/tls/private/serverX.key
...
```

5. Restart your web service. Note that a FAILED stop probably just indicates a service which was never originally started.

```
[root@serverX tls]# service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [ OK ]
```

6. (Optional) Use the **curl** command to confirm that clients can successfully verify encrypted connections with your server. You are not interested in the content, just that **curl** can download and verify the certificate without complaint.

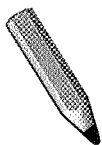
```
[root@serverX tls]# CACERT=/net/instructor/var/ftp/pub/example-ca.crt
[root@serverX tls]# curl --cacert $CACERT https://serverX.example.com > /dev/null
```

% Total	% Received	% Xferd	Average	Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left	Speed
102	3985	102	3985	0	0	62741	0	--:--:-- --:--:-- --:--:-- 1297k

7. In order to confirm that the Firefox web browser can successfully authenticate encrypted connections with your server, first install the local Certificate Authority's certificate by using Firefox to open the location <http://instructor/pub/example-ca.crt>.

In the resulting dialog, trust the CA to identify (at least) websites.

Browse <https://serverX.example.com>. Firefox should successfully verify the site using "Example, Inc.", which can be confirmed by "hovering" over the example.com prefix in the location bar.



Test

Criterion Test 2

Case Study

Web Server Additional Configuration

Before you begin...

Run the **lab-setup-website** script on desktopX.

Example Industries, a fine example of a company, needs a new website. In fact, they need two! One will be the company website and the other will be for testing content. Additionally, the company website will need a password protected area and a special CGI application installed.

On your serverX machine, deploy a web server with two virtual hosts.

Virtual host 1: *http://serverX.example.com*

- Create a simple placeholder page for the base URL

Virtual host 2: *http://wwwX.example.com*

- Create a simple placeholder page for the base URL that is different from the one used on virtual host 1
- Make *http://wwwX.example.com/private* a password protected area
- Add user **forrest** with password **trees** to **/private**
- Download the CGI file *ftp://instructor.example.com/pub/gls/special.cgi* and install it as *http://wwwX.example.com/cgi-bin/special.cgi*

When you are ready to check your work, run the grading script **lab-grade-website** on desktopX.

1. Create a directory which will serve as the DocumentRoot for your virtual website.

```
[root@serverX ~]# mkdir -p /var/www/virtual/wwwX/html
```

2. Create placeholder "home pages" for each of your sites by creating a distinctive **index.html** file in their respective DocumentRoots.

```
[root@serverX ~]# echo server > /var/www/html/index.html
[root@serverX ~]# echo www > /var/www/virtual/wwwX/html/index.html
```

3. Create the following minimal **/etc/httpd/conf.d/virtual.conf**, which contains virtual host definitions. Note that the filename does not matter, as long as it matches **/etc/httpd/conf.d/*.conf**.

```
NameVirtualHost 192.168.0.101

<VirtualHost 192.168.0.101>
    ServerName serverX.example.com
    ServerAlias serverX s1
</VirtualHost>

<VirtualHost 192.168.0.101>
    ServerName wwwX.example.com
    ServerAlias wwwX
    DocumentRoot /var/www/virtual/wwwX/html
</VirtualHost>
```

4. Restart your server, and confirm that the two virtual hosts serve the expected content.

```
[root@serverX ~]# service httpd restart
Stopping httpd:          [ OK ]
Starting httpd:          [ OK ]
[root@serverX ~]# curl http://serverX.example.com
server
[root@serverX ~]# curl http://wwwX.example.com
```

www

5. Create a private area for your virtual site, and create some test content for that area.

```
[root@serverX ~]# mkdir -p /var/www/virtual/wwwX/html/private
[root@serverX ~]# echo "ssshhhh" > /var/www/virtual/wwwX/html/private/secret
```

6. Add the following context stanza to your **virtual.conf** configuration file.

```
<Directory /var/www/virtual/wwwX/html/private>
    AuthName "Secret Hideout"
    AuthType basic

    AuthUserFile /etc/httpd/users
    require valid-user

    Options +Indexes
</Directory>
```

Note that enabling directory browsing was not specifically asked for, but mimics the "real server's" default behavior. More importantly, it will make the grading script happier.

7. Restart your server, and confirm that the context stanza has the anticipated effect.

```
[root@serverX ~]# service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [ OK ]
[root@serverX ~]# curl http://wwwX.example.com/private/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Authorization Required</title>
</head><body>
...
```

8. Create the specified web user and password.

```
[root@serverX ~]# htpasswd -cm /etc/httpd/users forrest
New password: trees
Re-type new password: trees
Adding password for user forrest
```

9. Confirm the specified user can access the private area.

```
[root@serverX ~]# curl http://forrest:trees@wwwX.example.com/private/secret
ssshhhh
```

10. Stretch and take a deep breath.

11. Inside **/etc/httpd/conf.d/virtual.conf**, add the following ScriptAlias directive inside your wwwX VirtualHost stanza.

```
<VirtualHost 192.168.0.101>
```

```
ServerName wwwX.example.com
ServerAlias wwwX
DocumentRoot /var/www/virtual/wwwX/html
ScriptAlias /cgi-bin/ "/var/www/virtual/wwwX/cgi-bin/"
</VirtualHost>
```

12. Restart your server, so the new configuration takes effect.

```
[root@serverX ~]# service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [ OK ]
```

13. Create the specified **cgi-bin** directory. Download the specified CGI executable, and install it into the newly created directory, and make it executable.

```
[root@serverX ~]# mkdir -p /var/www/virtual/wwwX/cgi-bin
[root@serverX ~]# curl http://instructor/pub/gls/special.cgi > /var/www/virtual/wwwX/
cgi-bin/special.cgi
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0   77    0   77    0     0  17416      0  --:--:-- --:--:-- --:--:--  77000
[root@serverX ~]# chmod 755 /var/www/virtual/wwwX/cgi-bin/special.cgi
```

14. Confirm that you can locally execute the CGI executable.

```
[root@serverX ~]# /var/www/virtual/wwwX/cgi-bin/special.cgi
Content-Type: text/html

<h1>Hello world</h1>
```

15. Confirm that the CGI executable can be accessed at the specified URL.

```
[root@serverX ~]# curl http://wwwX/cgi-bin/special.cgi
<h1>Hello world</h1>
```

16. Note: Because we chose "conforming" directory locations (locations which were registered in the SELinux policy) for our virtual host content, newly created files automatically obtained the correct SELinux context. Choosing different locations would require adjusting the SELinux types. The simplest approach would be to copy the SELinux types from the "known good" base server locations.

```
[root@serverX ~]# cd /var/www/virtual/wwwX/
[root@serverX wwwX]# chcon -R --reference /var/www/html html/
[root@serverX wwwX]# chcon -R --reference /var/www/cgi-bin cgi-bin/
[root@serverX wwwX]# ls -lZ
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
```

Basic SMTP Configuration



Practice Case Study

Intranet Configuration

Before you begin...

DNS has already been configured to overlay your hosts as members of the **domainX.example.com** domain.

hostname	ip address	also known as
mail.domainX.example.com	192.168.0.X+100	(serverX.example.com)
smtp.domainX.example.com	192.168.0.X+200	(hostX.example.com)
desktop.domainX.example.com	192.168.0.X	(desktopX.example.com)

Table A.1. domainX.example.com

Also, the host mail.domainX.example.com is the **MX** recipient for the entire domainX.example.com domain.

Complete the following table with the appropriate directives to configure these hosts to act as an intranet mailbox server, smtp host, and client station, respectively.

Try to use only the the **BASIC_CONFIGURATION_README**, **STANDARD_CONFIGURATION_README** and **main.cf** files for reference.

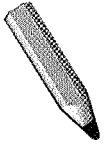
Once complete, have another student review your work.

External DNS

domainX.example.com. IN MX 10 mail.domainX.example.com.

Concept	Directive	mail.domainX	desktop.domainX	smtp.domainX
Binding Interface	<i>inet_interfaces</i>			
Masquerade as	<i>myorigin</i>			
Indirect Delivery	<i>relayhost</i>			
Receive mail for ...	<i>mydestination</i>			
Local Delivery	<i>local_transport</i>			
Relay from ...	<i>mynetworks</i>			

Table A.2. Intranet Mail Configuration for domainX.example.com



Test

Criterion Test

Case Study

Intranet E-mail Configuration

Before you begin...

Before you begin, run the script **lab-setup-email** on desktopX

The Hoffman Hair Supply company, a manufacturer of hair grooming products, wants to centralize the management of their internal e-mail.

DNS has already been configured so that your machines are members of the DNS domain **domainX.example.com** with the following addresses:

192.168.0.X	desktop.domainX.example.com	(a.k.a. desktopX.example.com)
192.168.0.X+100	mail.domainX.example.com	(a.k.a. serverX.example.com)
192.168.0.X+200	smtp.domainX.example.com	(a.k.a. hostX.example.com)

Also, the **mail.domainX.example.com** server is the **MX** recipient for the entire **domainX.example.com** domain.

Configure the **mail.domainX.example.com** host to act as an incoming mail-only server, so that all mail delivered to the **@domainX.example.com** domain is stored on this server.

Configure the **smtp.domainX.example.com** server to act as an outgoing SMTP server, which is willing to relay mail from members of the **domainX.example.com** domain to outside networks.

Configure the **desktop.domainX.example.com** host to act as a "null client". It cannot receive e-mail from the network, local mail delivery is disabled, and all outgoing e-mail is sent indirectly via **smtp.domainX.example.com**.

For all three hosts, make sure that any originating mail masquerades the sender's domain as **domainX.example.com**.

When you are finished run the **lab-grade-email** script to check your work.

1. These instructions will refer to your three machines in the lab's context, *desktop.domainX.example.com*, *mail.domainX.example.com*, and *smtp.domainX.example.com*, although the hostnames *desktopX*, *serverX*, and *hostX*, respectively, can still be used.

For brevity, these three machines will usually be referred to a *desktop*, *mail*, and *smtp*, with the *domainX.example.com* implied.

On *desktop.domainX.example.com* (*desktopX*), confirm that DNS has been properly pre-configured for your domain, by asking for a "domain dump" of *domainX.example.com*. (Note: in the "real world", most DNS servers will not allow clients to dump an entire domain.)

```
[root@desktop1 ~]# host -al domainX.example.com
...
;; QUESTION SECTION:
;domainX.example.com.      IN      AXFR
```



```
;; ANSWER SECTION:
domainX.example.com. 86400 IN SOA instructor.example.com.
root.instructor.example.com. 2009062000 3600 300 604800 60
domainX.example.com. 86400 IN NS instructor.example.com.
domainX.example.com. 86400 IN MX 10 mail.domainX.example.com.
desktop.domainX.example.com. 86400 IN A 192.168.0.X
mail.domainX.example.com. 86400 IN A 192.168.0.X+100
smtp.domainX.example.com. 86400 IN A 192.168.0.X+200
domainX.example.com. 86400 IN SOA instructor.example.com.
root.instructor.example.com. 2009062000 3600 300 604800 60
...
```

In particular, confirm the A records of *desktop*, *mail*, and *smtp* which perform the mapping to *domainX*, *serverX*, and *hostX*'s IP addresses, and the MX record that establishes *mail.domainX.example.com* as the "MX recipient" for the entire *domainX.example.com* domain. The rest of the DNS entries can be safely ignored.

2. In order to monitor your progress, it helps to monitor the `/var/log/maillog` on each of the three machines. As a suggestion, on the *desktop* machine, open 3 terminals. Leave one with a local shell, and open a remote shell to *mail* and *smtp* in each of the other two. In each of the terminals, run `less -F /var/log/messages`. (This places `less` in "tail -f" behavior, from which **CTRL-C** can be used to drop back into "normal" `less`, and **F** can be used to resume following the file.)
3. First, configure *mail* to be the MX recipient for the *domainX.example.com* domain. All of the following steps should be executed on *mail.domainX.example.com*.
 - a. Ensure that postfix is installed, started, and enabled.

```
[root@serverX ~]# yum -y install postfix
[root@serverX ~]# service postfix restart
[root@serverX ~]# chkconfig postfix on
```

- b. After making a backup of the primary configuration file, issue the following postfix configuration commands, and restart the server. (Of course, configuration changes can be made by editing the `/etc/postfix/main.cf` configuration file directly, which benefits from helpful comments.)

```
[root@serverX ~]# cp /etc/postfix/main.cf /etc/postfix/main.cf.orig
[root@serverX ~]# postconf -e inet_interfaces=all
[root@serverX ~]# postconf -e myorigin=domainX.example.com
[root@serverX ~]# postconf -e 'relayhost=[smtp.domainX.example.com]'
[root@serverX ~]# postconf -e mydestination=domainX.example.com
[root@serverX ~]# service postfix restart
```

- c. Confirm that mail sent locally to *student@domainX.example.com* is received by the host *mail*, which considers it a final destination.

```
[root@serverX ~]# date | mail -s test student@domainX.example.com
```

Examine the tail of `/var/log/maillog` on *mail* for lines similar to the following, where what is significant is "to=<student@domain1.example.com> ... (delivered to mailbox)".

```
...
Dec 10 05:39:48 serverX postfix/qmgr[28222]: 53948105A0:
  from=<root@serverX.example.com>, size=470, nrcpt=1 (queue active)
Dec 10 05:39:48 serverX postfix/local[28260]: 53948105A0:
  to=<student@domainX.example.com>, relay=local, delay=0.1,
  delays=0.07/0.02/0/0.01, dsn=2.0.0, status=sent (delivered to mailbox)
...
```

4. Secondly, configure *smtp* to be an outgoing mail relay. All of the following steps should be executed on *smtp.domainX.example.com*.

- a. Ensure that postfix is installed, started, and enabled.

```
[root@hostX ~]# yum -y install postfix
[root@hostX ~]# service postfix restart
[root@hostX ~]# chkconfig postfix on
```

- b. After making a backup of the primary configuration file, issue the following postfix configuration commands, and restart the server.

```
[root@hostX ~]# cp /etc/postfix/main.cf /etc/postfix/main.cf.orig
[root@hostX ~]# postconf -e inet_interfaces=all
[root@hostX ~]# postconf -e myorigin=domainX.example.com
[root@hostX ~]# postconf -e local_transport="error:local delivery disabled"
[root@hostX ~]# postconf -e mynetworks="127.0.0.0/8 192.168.0.0/24"
[root@hostX ~]# service postfix restart
```

- c. Unfortunately, at this point, there's little that can be confirmed, other than local email delivery has been disabled.

```
[root@hostX ~]# date | mail -s test student
```

Examine the tail of */var/log/maillog* on *smtp* for lines similar to the following, where what is significant is "status=bounced (local delivery disabled)"

```
...
Dec 10 05:57:44 hostX postfix/bounce[27732]: 49BCD10591: sender non-delivery
  notification: 657C210594
Dec 10 05:57:44 hostX postfix/qmgr[27724]: 49BCD10591: removed
Dec 10 05:57:44 hostX postfix/error[27731]: 657C210594:
  to=<root@hostX.example.com>, relay=none, delay=0.02, delays=0.01/0/0/0,
  dsn=5.0.0, status=bounced (local delivery disabled)
...
```

5. Thirdly, configure *desktop* to be a null client. All of the following commands are to be run on *desktop.domainX.example.com*.

- a. Ensure that postfix is installed, started, and enabled.

```
[root@desktopX ~]# yum install postfix
[root@desktopX ~]# service postfix restart
[root@desktopX ~]# chkconfig postfix on
```

- b. After making a backup of the primary configuration file, issue the following postfix configuration commands, and restart the server.

```
[root@desktopX ~]# cp /etc/postfix/main.cf /etc/postfix/main.cf.orig
[root@desktopX ~]# postconf -e myorigin=domainX.example.com
[root@desktopX ~]# postconf -e 'relayhost=[smtp.domainX.example.com]'
[root@desktopX ~]# postconf -e local_transport="error:local delivery disabled"
[root@desktopX ~]# service postfix restart
```

6. As a final confirmation of your work, from *desktop*, send a test message to `student@domainX.example.com`. Through the various `/var/log/message` files, trace the path of the email as it originates from *desktop*, passes through *smtp* as an outgoing relay, and is finally received by *mail* as its final destination.

```
[root@desktopX ~]# date | mail -s final_test student@domainX.example.com
```

- a. On *desktop*, examine the tail of `/var/log/maillog` for lines similar to the following, where what is significant is `'relay="smtp.domainX.example.com[192.168.0.X+200]:25"'` and `'status=sent'`.

```
Dec 10 14:32:04 desktopX postfix/qmgr[8860]: 7D35D247D7:
  from=<root@desktopX.example.com>, size=473, nrcpt=1 (queue active)
Dec 10 14:32:04 desktopX postfix/smtp[8905]: 7D35D247D7:
  to=<student@domainX.example.com>, relay=smtp.domainX.example.com[192.168.0.X
+200]:25, delay=0.18, delays=0.06/0.02/0.06/0.04, dsn=2.0.0, status=sent (250
2.0.0 Ok: queued as 606C510594)
Dec 10 14:32:04 desktopX postfix/qmgr[8860]: 7D35D247D7: removed
```

- b. On *smtp*, examine the tail of `/var/log/maillog` for lines similar to the following, where both the reception (`"client=desktopX.example.com[192.168.0.X] ... from=<root@desktopX.example.com>"`) and the subsequent transmission (`"to=<student@domainX.example.com> ... relay=mail.domainX.example.com[192.168.0.X+100]:25, ... status=sent"`) of the email should be evident.

```
...
Dec 10 06:20:02 hostX postfix/smtpd[27799]: connect from
desktopX.example.com[192.168.0.X]
Dec 10 06:20:02 hostX postfix/smtpd[27799]: 606C510594:
  client=desktopX.example.com[192.168.0.X]
Dec 10 06:20:02 hostX postfix/cleanup[27802]: 606C510594: message-
id=<20101210193204.7D35D247D7@desktopX.example.com>
Dec 10 06:20:02 hostX postfix/qmgr[27724]: 606C510594:
  from=<root@desktopX.example.com>, size=684, nrcpt=1 (queue active)
Dec 10 06:20:02 hostX postfix/smtpd[27799]: disconnect from
desktopX.example.com[192.168.0.X]
Dec 10 06:20:02 hostX postfix/smtp[27803]: 606C510594:
  to=<student@domainX.example.com>, relay=mail.domainX.example.com[192.168.0.X
+100]:25, delay=0.13, delays=0.02/0.02/0.05/0.04, dsn=2.0.0, status=sent (250
2.0.0 Ok: queued as 7EA04105A0)
...
```

- c. On *mail*, examine the tail of `/var/log/maillog` for lines similar to the following, where the reception (`client=hostX.example.com[192.168.0.X`

+200] ... from=<root@desktopX.example.com> and final delivery of the email ("to=<student@domainX.example.com> ... status=sent (delivered to mailbox)") should be evident.

```
...
Dec 10 06:20:02 serverX postfix/cleanup[28475]: 7EA04105A0: message-
id=<20101210193204.7D35D247D7@desktopX.example.com>
Dec 10 06:20:02 serverX postfix/qmgr[28457]: 7EA04105A0:
  from=<root@desktopX.example.com>, size=897, nrcpt=1 (queue active)
Dec 10 06:20:02 serverX postfix/smtpd[28472]: disconnect from
  hostX.example.com[192.168.0.X+200]
Dec 10 06:20:02 serverX postfix/local[28476]: 7EA04105A0:
  to=<student@domainX.example.com>, relay=local, delay=0.05,
  delays=0.02/0.02/0/0.01, dsn=2.0.0, status=sent (delivered to mailbox)
Dec 10 06:20:02 serverX postfix/qmgr[28457]: 7EA04105A0: removed
...
```

- d. Lastly, as the user *student* on *mail*, check your email using your favorite email client, such as **mutt**.

```
[root@serverX ~]# yum install -y mutt
[root@serverX ~]# su - student
[student@serverX ~]# mutt
```

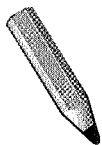
By listing *all* headers (within **mutt**, this requires typing **h** while viewing the message), you should be able to trace the path of the email in reverse order.

```
...
Received: from hostX.example.com (hostX.example.com [192.168.0.X+200])
  by serverX.example.com (Postfix) with ESMTP id 7EA04105A0
  for <student@domainX.example.com>; Fri, 10 Dec 2010 06:20:02 -0500
  (EST)
Received: from desktopX.example.com (desktopX.example.com [192.168.0.X])
  by hostX.example.com (Postfix) with ESMTP id 606C510594
  for <student@domainX.example.com>; Fri, 10 Dec 2010 06:20:02 -0500
  (EST)
Received: by desktopX.example.com (Postfix, from userid 0)
  id 7D35D247D7; Fri, 10 Dec 2010 14:32:04 -0500 (EST)
...
```

7. Lastly, from *desktop*, let your instructor in on the fun by sending mail to *instructor.example.com*.

```
[root@desktopX ~]# echo whew | mail -s done instructor@instructor.example.com
```

Caching-Only DNS Server



Test

Criterion Test

Case Study

Caching-Only DNS Server

Before you begin...

Before you begin, run the script **lab-setup-cachingdns** on desktopX.

For his growing import/export business, Mr. Hnath would like to improve name resolution performance by deploying a caching name server at each of his business locations.

Recursive queries should be forwarded to the main name server at Hnath Import/Export headquarters.

- Set up a caching name server on serverX.
- Configure the name server so that recursive queries are sent to **instructor.example.com**. Also, configure the name server to accept queries from anyone on the classroom network.

When you are ready, run **lab-grade-cachingdns** on desktopX to check your work.

1. Make sure the bind package is installed.

```
[root@serverX ~]# yum install bind
```

2. Modify the named configuration (**/etc/named.conf**) to support connections from the network by modifying the **listen-on** lines to look like the following.

```
listen-on port 53 { any; };
listen-on-v6 port 53 { any; };
```

3. Modify the named configuration (**/etc/named.conf**) to ignore DNSSEC by modifying the **dnssec-validation** line to look like the following.

```
dnssec-validation no;
```

4. Modify the named configuration (**/etc/named.conf**) to accept queries from anyone on the classroom network by modifying the **allow-query** line to look like the following. Also, modify the configuration so that recursive queries are sent to **instructor.example.com** by inserting a **forwarders** line below the **allow-query** line as the following.

```
allow-query { localhost; 192.168.0.0/24; };
forwarders { 192.168.0.254; };
```

5. Restart the named service. Note that a FAILED stop probably just indicates a service which was never originally started.

```
[root@server1 ~]# service named restart
Stopping named:          [ OK ]
Starting named:          [ OK ]
```

6. Test from the desktopX system.

```
[root@desktop1 ~]# host server1.example.com 192.168.0.101
Using domain server:
Name: 192.168.0.101
Address: 192.168.0.101#53
Aliases:

server1.example.com has address 192.168.0.101
```

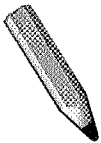
File Sharing with NFS



Practice Quiz

NFS Concepts Quiz

1. Under what circumstances should NFSv2 or NFSv3 be used? Legacy installations and some network clients don't support NFSv4
2. What is the syntax of the /etc/exports file? /directory/share host(options) host(options)
3. What steps should be taken to publish a new export on an existing NFSv4 server?
Create an /etc/exports entry that defines access to the share
Run **exportfs -r** as root on the server
4. Which option tells NFS to allow the root user on client systems to have root privileges in the share as well?
no_root_squash



Test

Criterion Test

Case Study

File Sharing with NFS

Before you begin...

Make sure to run **lab-setup-strickland** from your desktopX system, which will prepare your serverX system for the lab.

Strickland Pro Play is a store specializing in high end recreational equipment and accessories. The new sales software requires requires a file server with two shares that are mounted at each sales station in the store.

For the file server, deploy an NFSv4 service on desktopX. Create and share two exports on desktopX:

- The first export is for intake of current sales orders. On desktopX, export **/share/current** and make it writable. Root on the client must be able to write to **/share/current** when mounted. The second export is for order archives.
- The second export is to archive old orders. Again on desktopX, export the path **/share/archives** and make it read-only.
- Configure both exports so they are only available to the local classroom network.

Configure serverX to mount **desktopX:/share/current** as **/sales/current** and **desktopX:/share/archives** as **/sales/archives**. The mounts must be available after a reboot of serverX.

When you are ready, run the **lab-grade-strickland** script on serverX to check your work.

1. First, create both directories to be shared on the NFS server, desktopX. Make the **current** directory writeable.

```
[root@desktopX ~]# mkdir -p /share/archives /share/current
[root@desktopX ~]# chmod 777 /share/current
```

(Another acceptable option is to export **/share/current** with the **no_root_squash** option set, in the next step.)

2. Define how both directories will be shared by creating **/etc/exports**.

```
[root@desktopX ~]# vi /etc/exports
[root@desktopX ~]# cat /etc/exports
/share/current 192.168.0.0/24(rw, sync)
/share/archives 192.168.0.0/24(ro, async)
```

3. Start the NFS service and configure it to start persistently.

```
[root@desktopX ~]# service nfs start
Starting NFS services:           [ OK ]
Starting NFS quotas:            [ OK ]
Starting NFS daemon:            [ OK ]
Starting NFS mountd:            [ OK ]
[root@desktopX ~]# chkconfig nfs on
```

4. Once the NFS server is configured and running, login as root on serverX and configure the NFS client. First, create the mountpoints.

```
[root@serverX ~]# mkdir -p /sales/archives /sales/current
```

5. Confirm the NFS shares exported by desktopX.example.com are visible to serverX and create **/etc/fstab** entries at the end to mount them.

```
[root@serverX ~]# showmount -e desktopX.example.com
Export list for desktopX.example.com:
/share/archives 192.168.0.0/24
/share/current 192.168.0.0/24
[root@serverX ~]# vi /etc/fstab
[root@serverX ~]# tail -n 2 /etc/fstab
desktopX.example.com:/share/current /sales/current nfs rw 0 0
desktopX.example.com:/share/archives /sales/archives nfs ro 0 0
```

6. Mount the NFS shares and confirm they mounted correctly. If all is well, then reboot serverX and run the lab-grade-strickland grading script.

```
[root@serverX ~]# mount -a
[root@serverX ~]# mount | grep desktop
```



```
desktopX.example.com:/share/current on /sales/current type nfs
(rw,vers=4,addr=192.168.0.1,clientaddr=192.168.0.101)
desktopX.example.com:/share/archives on /sales/archives type nfs
(ro,vers=4,addr=192.168.0.1,clientaddr=192.168.0.101)
[root@serverX ~]# reboot
```

File Sharing with CIFS



Practice Quiz

Accessing CIFS Share Quiz

1. What command-line would give ftp-style access to a CIFS share named "common" on a server named "nas2010", logging you in as a user named "winston"?

smbclient //nas2010/common -U winston

2. What is wrong with the following line in /etc/fstab?

\\server\share /mnt/point cifs user=ralph,pass=password 0 0

Trick question: there is actually nothing wrong with this line. You can either use backslashes or forward slashes in /etc/fstab. However, on the command-line, you must double the backslashes if you choose to use them: **smbclient \\serverX\\share**

3. How would you store the login credentials in a separate file to keep them out of /etc/fstab?

in **fstab** (column 4): **credentials=filename** and **filename** contains on three lines: **username=value**, **password=value**, and **domain=value**

4. When mounting a Windows-based CIFS share, what option allows you to specify the Linux ownership of all mounted files?

uid=value, **gid=value** where value can be either the Linux UID/GID or username/groupname



Practice Performance Checklist

Samba Home Directories Configuration Exercise

Modify the default Samba configuration and security elements to support access to user home directories.

- ☐ Log into serverX and escalate privileges to root

[student@serverX ~]# su -

- ☐ Install the necessary package(s) for a Samba server

[root@serverX ~]# yum install -y samba

- ☐ Start and enable the Samba service

[root@serverX ~]# service smb start
[root@serverX ~]# chkconfig smb on

- ☐ Configure system to be in the CLASSX workgroup (where X is your station number) with local user definitions.

Edit `/etc/samba/smb.conf` and set the following fields.

```
workgroup = CLASSX # X is your desktop number
security = user      # default
passdb backend = tdbsam # default
```

- ☐ Add a Samba-only user named **winuserX** (where X is your station number) with a Samba password of **winpass**.

```
[root@serverX ~]# useradd -s /sbin/nologin winuserX
[root@serverX ~]# smbpasswd -a winuserX
New SMB password: winpass
Retype new SMB password: winpass
...
Added user winuserX.
```

- ☐ Enable user home directory access in SELinux

```
[root@serverX ~]# setsebool -P samba_enable_home_dirs on
```

- ☐ Enable the firewall and open up necessary ports to grant access.

```
[root@serverX ~]# iptables -I INPUT -p udp --dport 137:138 -j ACCEPT
[root@serverX ~]# iptables -I INPUT -p tcp --dport 139 -j ACCEPT
[root@serverX ~]# iptables -I INPUT -p tcp --dport 445 -j ACCEPT
[root@serverX ~]# service iptables save
```

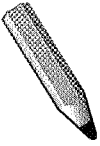
- ☐ Test the configuration, by accessing your Samba-only user's home directory from desktopX.

```
[root@serverX ~]# smbclient //serverX/winuserX -U winuserX%winpass
```

To test using the GUI, go to **Places → Connect to Server**. Fill in the following fields (leave the others blank and remember to substitute your desktop number for X):

```
Service type: Windows share
Server: serverX
Share: winuserX
User Name: winuserX
Domain Name: CLASSX
```

When prompted, enter **winpass** as the password.



Test

Criterion Test

Case Study

File Sharing with CIFS

Before you begin...

Make sure to run the **lab-setup-samba** from your desktopX system, which will prepare your serverX system for the lab.

The School of Butler and Hacker has recently deployed several CIFS servers to allow their Windows client systems access to file shares.

The Color Guard, known as Green and Red is deploying a new server and needs to share information using CIFS. That share must be writable by members of the Color Guard, but other people can only have read access.

Enable the firewall and allow all clients on the local network to access the CIFS server.

Configure your serverX to function as a CIFS server, with the following information:

- Workgroup: BUTLER
- Linux Group: greenred
- CIFS Share Name: school
- Directory: /shared/school
- No printers shared

Test the configuration by:

- Creating a user as a member of **greenred** and ensuring they can write to the CIFS share, **school**
- instructor.example.com provides several printers that CUPS should automatically enable. Before you fulfill the printer requirement, check to verify they are available (they should be named printerX). Configure Samba so that no printers are shared and confirm that the user can NOT see them listed with **smbclient**
- Creating a second user not as a member of **greenred** and ensuring they can only read from the CIFS share, **school**

When you are ready, run the **lab-grade-samba** script on serverX to check your work.

1. Install, start and enable needed packages.

```
[root@server1 ~]# yum install samba samba-doc
[root@server1 ~]# service smb start
[root@server1 ~]# chkconfig smb on
```

2. Enable the firewall and add the following rules.

```
[root@server1 ~]# iptables -A INPUT -p udp --dport 137:138 -j ACCEPT
[root@server1 ~]# iptables -A INPUT -p tcp --dport 139 -j ACCEPT
[root@server1 ~]# iptables -A INPUT -p tcp --dport 445 -j ACCEPT
[root@server1 ~]# service iptables save
```

3. Create the Linux group and directory and configure SELinux support.

```
[root@server1 ~]# groupadd -r greenred
[root@server1 ~]# mkdir -p /shared/school
[root@server1 ~]# chgrp greenred /shared/school
[root@server1 ~]# chmod 2775 /shared/school
[root@server1 ~]# semanage fcontext -a -t public_content_t '/shared(/.*)?'
[root@server1 ~]# semanage fcontext -a -t samba_share_t '/shared/school(/.*)?'
[root@server1 ~]# restorecon -vvFR /shared
restorecon reset /shared context unconfined_u:object_r:default_t:s0-
>system_u:object_r:public_content_t:s0
restorecon reset /shared/school context unconfined_u:object_r:default_t:s0-
>system_u:object_r:samba_share_t:s0
```

4. Edit `/etc/samba/samba.conf`.

Modify or confirm the following:

```
[global]
...
workgroup = BUTLER
...
security = user
passdb backend = tdbsam
```

Add a section (at the end of the file) as follows.

```
[school]
path = /shared/school
write list = @greenred
read only = yes
guest ok = no
```

Comment or remove the following lines, for printers.

```
[global]
...
load printers = no
-OR-
#[printers]
#   comment = All Printers
#   path = /var/spool/samba
#   browseable = no
#   guest ok = no
#   writable = no
#   printable = yes
```

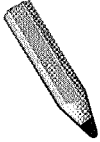
5. Restart samba.

```
[root@serverX]# service smb restart
```

6. Test as mentioned above.

```
[root@server1 ~]# useradd -s /sbin/nologin -G greenred red
[root@server1 ~]# smbpasswd -a red
New SMB password: red
Retype new SMB password: red
Added user red.
[root@server1 ~]# useradd -s /sbin/nologin bernice
[root@server1 ~]# smbpasswd -a bernice
New SMB password: bernice
Retype new SMB password: bernice
Added user bernice.
[root@server1 ~]# smbclient -L serverX -U red%red | grep printer
Domain=[BUTLER] OS=[Unix] Server=[Samba 3.5.4-68.el6]
Domain=[BUTLER] OS=[Unix] Server=[Samba 3.5.4-68.el6]
[root@server1 ~]# smbclient //serverX/school -U red%red
smb: \> put /etc/hosts hosts
smb: \> ls
        hosts                                A          177  Tue Dec 21 10:10:03 2010
smb: \> exit
[root@server1 ~]# smbclient //serverX/school -U bernice%bernice
smb: \> mkdir test
NT_STATUS_MEDIA_WRITE_PROTECTED making remote directory \test
smb: \> get hosts
getting file \hosts of size 177 as hosts (57.6 KiloBytes/sec) (average 57.6 KiloBytes/
sec)
```

File Sharing with FTP



Test

Criterion Test

Case Study

FTP Drop Box

Before you begin...

Make sure to run the **lab-setup-dropbox** from your desktopX system, which will prepare your serverX system for the lab.

The Quiet Pleases company, a manufacturer of silence cones and other noise canceling devices, has a program to collect information about noise levels around the world. Volunteers have been collecting data about noise and need an easy way to send in reports.

The company has decided to use an FTP server with an anonymous upload directory to collect the reports.

Deploy vsftpd on your serverX and configure a write-only upload directory that is accessible at: `ftp://serverX.example.com/dropbox`

As the volunteers are located all over world, the FTP server must accept connections from anywhere on the internet.

When you are ready, run the **lab-grade-dropbox** script on desktopX to check your work.

1. Install, start and enable needed packages.

```
[root@server1 ~]# yum install vsftpd
[root@server1 ~]# service vsftpd start
[root@server1 ~]# chkconfig vsftpd on
```

2. Create upload directory.

```
[root@server1 ~]# mkdir /var/ftp/dropbox
[root@server1 ~]# chgrp ftp /var/ftp/dropbox
[root@server1 ~]# chmod 730 /var/ftp/dropbox
```

3. Configure SELinux support.

```
[root@server1 ~]# semanage fcontext -a -t public_content_rw_t '/var/ftp/dropbox(/.*)'
[root@server1 ~]# restorecon -vvFR /var/ftp/dropbox
restorecon reset /var/ftp/dropbox context unconfined_u:object_r:public_content_t:s0-
>system_u:object_r:public_content_rw_t:s0
[root@server1 ~]# setsebool -P allow_ftpd_anon_write on
```

4. Edit `/etc/vsftpd/vsftpd.conf`.

Modify, uncomment or confirm the following:

```
anon_upload_enable=yes
chown_uploads=yes
chown_username=daemon
anon_umask=077
```

Restart **vsftpd**.

```
[root@server1 ~]# service vsftpd restart
Shutting down vsftpd: [ OK ]
Starting vsftpd for vsftpd: [ OK ]
```

5. Configure iptables (if enabled).

Edit **/etc/sysconfig/iptables-config**:

```
IPTABLES_MODULES="nf_conntrack_ftp nf_nat_ftp"
```

Edit **/etc/sysconfig/iptables**:

```
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p tcp --dport 21 -j ACCEPT
```

Restart **iptables**.

```
[root@server1 ~]# service iptables restart
```


CUPS Service



Practice Group Exercise

Manage Print Queues

1. Create a local print queue and share it with other systems. Name the print queue **local** and make it a text-only printer that points to either the serial or parallel port on your system.



Note

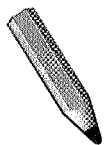
A text-only printer will not accept PostScript files like that sent by the **Print Test** feature. Do not be alarmed that the test page will not print.

2. Create a second print queue that points to a partner's local print queue. Name the print queue **remote** and make it a raw print queue that forwards jobs to your partner's **local** print queue.
3. When you finish, print some text files to **local** and **remote** to verify.



Note

If you are using a serial port, the print jobs are sent to the serial port almost immediately, so it may be difficult to verify that your print queues are working properly. If this is the case, use the "count" files (e.g., c00001, c00002, etc.) in **/var/spool/cups/** to verify. There will be a new count file created every time a print job goes through the queue.



Test

Criterion Test

Exercise

Configure and Manage a Printer

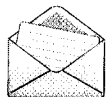
Before you begin...

From desktopX, run **lab-setup-cups** to reset your virtual server for this lab.

1. Configure a network printer to send print jobs to an IPP print queue on instructor.example.com called /printers/printerX where X is your desktop number.

```
[root@desktopX ~]# system-config-printer
```

Click **New**. Expand **Network Printer**. Select **Internet Printing Protocol (ipp)** and enter **instructor.example.com** as the **Host** and **/printers/printerX** as the **Queue**. Alternately, you could select **Find Network Printer** and enter **instructor.example.com** as the hostname, then click **Find**. Make sure **/printers/printerX** is in the queue, then click **Verify**.



Note

You must enter a fully-qualified domain name when searching for a network printer, or CUPS may not find it.

Once you have entered and verified the printer, click **Forward**.

2. Your print queue should be called **remote-test** and should be the default print queue.

Choose **Generic** as the printer and click **Forward**. Choose **text-only printer** as the model and click **Forward**. Enter **remote-test** as the **Printer Name** and click **Forward**.

If **remote-printer** is not the default, right-click on **remote-test** and choose **Set As Default**.

3. When you finish, run the evaluation script, **lab-grade-cups**.

```
[root@serverX ~]# lab-grade-cups
```

SSH Service



Practice Exercise

Using SSH Keys

1. Create an SSH key pair as **student** on desktopX.

```
[student@desktopX ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): Enter
Enter passphrase (empty for no passphrase): redhat
Enter same passphrase again: redhat
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
dc:cb:9b:30:98:b4:99:19:fc:fe:ba:97:2b:95:4d:29 student@desktop1.example.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|       .
|      .E o
|     + S . =
|    . X . o .
|   B + . o .
|  ..ooo
|  +* =
+-----+
```

2. Install the SSH public key for the **student** account on serverX.

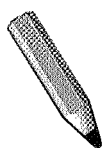
```
[student@desktopX ~]$ ssh-copy-id serverX
student@serverX's password: student
Now try logging into the machine, with "ssh 'serverX'", and check in:

    .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

3. Connect to serverX from desktopX using the SSH keys.

```
[student@desktopX ~]$ ssh serverX
Enter passphrase for key '/home/student/.ssh/id_rsa': redhat
[student@serverX ~]$
```



Test

Criterion Test

Exercise

Securing SSH

Before you begin...

Run the **lab-setup-server** command as **root** on your desktopX system. This will prepare your serverX system for the lab.

1. Copy the SSH public key generated previously on desktopX to the **student** account on serverX.

```
[student@desktopX ~]$ ssh-copy-id -i .ssh/id_rsa.pub student@serverX
```

2. Confirm you can **ssh** into serverX as **student** from desktopX using the SSH keys.

```
[student@desktopX ~]$ ssh student@serverX  
[student@serverX ~]$
```

Virtual Network Computing (VNC) Service



Practice Exercise

Enabling a VNC Server

1. Install the **tigervnc-server** package on serverX.

```
[root@serverX ~]# yum install tigervnc-server
```

2. Configure VNC display 1 for student. Add the following to **/etc/sysconfig/vncservers**:

```
VNCSERVERS="1:student"
```

3. Set **redhat** as the VNC password for student:

```
[student@serverX ~] vncpasswd  
Password: redhat  
Verify: redhat
```

4. Start and enable the VNC service.

```
[root@serverX ~]# service tigervnc start  
[root@serverX ~]# chkconfig tigervnc on
```

5. You will verify the connection in the next section.



Practice Exercise

Connect to VNC securely

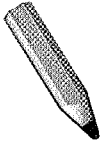
1. Configure the VNC server on serverX to allow local connections only. Edit **/etc/sysconfig/vncservers** and add the following:

```
VNCSERVERARGS[1]="-localhost"
```

2. Connect to the VNC server on serverX securely from desktopX using an SSH tunnel:

```
[student@desktopX ~] vncviewer -via serverX localhost:1
```

3. Verify everything is completed as specified.



Test

Criterion Test

Exercise

Configure Multiple Desktops with VNC

Before you begin...

Run the **lab-setup-server** command as **root** on your desktopX system. This will prepare your serverX system for the lab.

1. Install the VNC server package on serverX.

```
[root@serverX ~]# yum -y install tigervnc-server
```

2. Configure display 1 for **student** and display 2 for **visitor**.

Add the following to **/etc/sysconfig/vncservers**

```
VNCSEVERARGS="1:student 2:visitor"
```

3. Only permit connections from localhost.

Add the following to **/etc/sysconfig/vncservers**

```
VNCSEVERARGS[1]="-localhost"
VNCSEVERARGS[2]="-localhost"
```

4. Set **redhat** as the VNC passwords for both **student** and **visitor**.

```
[root@serverX ~]# su - student
[student@serverX ~]$ vncpasswd
Password: redhat
Verify: redhat
[student@serverX ~]$ exit
[root@serverX ~]# su - visitor
[visitor@serverX ~]$ vncpasswd
Password: redhat
Verify: redhat
[visitor@serverX ~]$ exit
```

5. Start and enable the VNC service.

```
[root@serverX ~]# service vncserver start
[root@serverX ~]# chkconfig vncserver on
```



Note

You must set the VNC passwords for each user before starting the service. Otherwise, the **vncserver** service will not start properly.

6. Verify everything is completed as specified, then check your work using a secure connection.

```
[root@desktopX ~]# vncviewer -via student@serverX localhost:1
TigerVNC Viewer for X version 1.0.90 - built Jun 30 2010 11:30:49
Copyright (C) 2002-2005 RealVNC Ltd.
Copyright (C) 2000-2006 TightVNC Group
Copyright (C) 2004-2009 Peter Astrand for Cendio AB
See http://www.tigervnc.org for information on TigerVNC.
The authenticity of host 'serverX (192.168.0.X+100)' can't be established.
RSA key fingerprint is 5c:77:a4:e3:23:9e:72:cf:ac:d4:cd:a7:6b:c4:94:ba.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'serverX' (RSA) to the list of known hosts.
student@serverX's password: student
```

```
VNC authentication
Password: redhat
```

```
[root@desktopX ~]# vncviewer -via visitor@serverX localhost:2
TigerVNC Viewer for X version 1.0.90 - built Jun 30 2010 11:30:49
Copyright (C) 2002-2005 RealVNC Ltd.
Copyright (C) 2000-2006 TightVNC Group
Copyright (C) 2004-2009 Peter Astrand for Cendio AB
See http://www.tigervnc.org for information on TigerVNC.
visitor@serverX's password: password
```

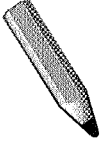
```
VNC authentication
Password: redhat
```



Note

The parameter after **-via** is used to connect using **ssh**. It is not necessary to use the username to whose VNC session you are connecting. Any username would work, as long as you know the password.

Comprehensive Review



Test

Comprehensive Review Test

Exercise

Comprehensive Review

Before you begin...

Run the **lab-setup-server** command as **root** on desktopX.

Configure serverX so that it meets the following requirements. For all services, allow connections from the local 192.168.0.0/24 subnet, but disallow connections from the 192.168.1.0/255.255.255.0 subnet.

1. Configure SELinux to run in Enforcing mode.

Ensure **/etc/sysconfig/selinux** contains:

```
SELINUX=enforcing
```

2. Allow SSH connections from the local subnet.

```
[root@serverX ~]# iptables -I INPUT -m state --state NEW -s 192.168.0.0/24 -p tcp --
dport 22 -j ACCEPT
[root@serverX ~]# iptables -I INPUT -i lo -j ACCEPT
[root@serverX ~]# iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
[root@serverX ~]# iptables -A INPUT -m state --state NEW -j REJECT
[root@serverX ~]# service iptables save
```

3. Configure an SMTP server that allows connections from the local subnet.

Edit **/etc/postfix/main.cf** and change:

```
inet_interfaces = localhost
```

to:

```
inet_interfaces = all
```

```
[root@serverX ~]# service postfix restart
[root@serverX ~]# iptables -I INPUT 3 -m state --state NEW -s 192.168.0.0/24 -p tcp --
dport 25 -j ACCEPT
[root@serverX ~]# service iptables save
```




Note

We want the **ESTABLISHED, RELATED** and loopback, (**-i lo**) rules as our first rules. We do not want to place this new rule at the end (**-A**) because it would come after the **REJECT** rule, so **-I INPUT 3** places this rule as the third rule.

4. Connect to the LDAP server, `instructor.example.com`, using the distinguished name (DN) of **dc=example, dc=com** for account information. The LDAP server requires secure connections using the certificate found at `ftp://instructor.example.com/pub/EXAMPLE-CA-CERT`. The LDAP server provides an account named **ldapuserX**.

Use Kerberos passwords with a realm **EXAMPLE.COM** for authentication. Set the KDC and Admin servers to `instructor.example.com`. The accounts have a password of **kerberos**.

For the **system-config-authentication** GUI, choose **LDAP** in the **User Account Database** drop-down menu. Change the **LDAP Server** to **ldap://instructor.example.com**. Select **Use TLS to encrypt connections**. Click on the **Download CA Certificate...** button and enter `ftp://instructor/pub/EXAMPLE-CA-CERT`. Change the **KDCs** and **Admin Servers** to **instructor.example.com**. Leave the other settings as they are and click **Apply**.

To use the command-line tool **authconfig**, use the following command:

```
authconfig --enableldap --ldapserver=instructor.example.com --enableldaptls \
--ldaploadcacert=ftp://instructor.example.com/pub/EXAMPLE-CA-CERT \
--ldapbasedn="dc=example,dc=com" --disableldapauth --enablekrb5 \
--krb5kdc=instructor.example.com --krb5adminserver=instructor.example.com \
--krb5realm=EXAMPLE.COM --enablesssd --enablesssdauth --update
```

5. Configure an automounted home directory for the **ldapuserX** account. The home directory is shared via NFS from `instructor.example.com`.

Add the following to the **/etc/auto.master** file:

```
/home/guests /etc/auto.guests
```

Create **/etc/auto.guests** and add the following content:

```
* instructor.example.com:/home/guests/&
```

```
[root@serverX ~]# service autofs reload
```

6. Connect to the iSCSI target **rdisks.serverX** provided by `instructor.example.com`.

```
[root@serverX ~]# iscsiadm -m discovery -t st -p 192.168.0.254
[root@serverX ~]# iscsiadm -m node -T iqn.2010-09.com.example:rdisks.serverX -p
192.168.0.254 -l
```

- Remove all of the current partitions on the iSCSI disk. Configure a new 30 MB physical partition using the iSCSI target with an ext4 filesystem and a label of **test** mounted on **/test/**. The **/test/** directory must be owned by the user **root** and the group **root**, and have a permission of 755.

```
[root@serverX ~]# dd if=/dev/zero of=/dev/sda count=1
[root@serverX ~]# fdisk -cu /dev/sda
Command (m for help): n
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (2048-65535, default 2048): Enter
Last sector, +sectors or +size{K,M,G} (2048-65535, default 65535): +30MB
Command (m for help): w
[root@serverX ~]# mkfs -t ext4 -L test /dev/sda1
[root@serverX ~]# mkdir /test
```

Add a line to the **/etc/fstab** file:

```
LABEL=test      /test  ext4    _netdev 1 2
```

```
[root@serverX ~]# mount -a
[root@serverX ~]# chown root:root /test
[root@serverX ~]# chmod 755 /test
```

- Configure a new 1 GB logical volume named **mylv** in the **vgssrv** volume group, with an ext4 filesystem mounted on **/mylv/**.



Note

vgs shows that we do not have enough space to create a 1 GB logical volume, so we must first add some disk space to the volume group.

```
[root@serverX ~]# fdisk -cu /dev/vda
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (9914368-12582911, default 9914368): Enter
Using default value 9914368
Last sector, +sectors or +size{K,M,G} (9914368-12582911, default 12582911): +1G

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
Changed system type of partition 3 to 8e (Linux LVM)

Command (m for help): w
[root@serverX ~]# reboot
```

```
[root@serverX ~]# pvcreate /dev/vda3
[root@serverX ~]# vgextend vgsrv /dev/vda3
[root@serverX ~]# lvcreate -L 1G -n mylv vgsrv
[root@serverX ~]# mkfs -t ext4 /dev/vgsrv/mylv
[root@serverX ~]# mkdir /mylv
```

Add the following line to the **/etc/fstab** file:

```
/dev/mapper/vgsrv-mylv /mylv ext4 defaults 1 2
```

```
[root@serverX ~]# mount -a
```

9. Configure NFS to share the **/test/** directory. Make it read-only to the local subnet. Allow **root** to have root privileges when accessing the NFS share.

Add the following line to the **/etc/exports** file:

```
/test 192.168.0.0/24(ro,no_root_squash)
```

```
[root@serverX ~]# service nfs restart
[root@serverX ~]# chkconfig nfs on
[root@serverX ~]# iptables -I INPUT 3 -m state --state NEW -s 192.168.0.0/24 -p tcp --
dport 2049 -j ACCEPT
[root@serverX ~]# service iptables save
```

10. Create a user account named **matt** using a password of **matt**.

```
[root@serverX ~]# useradd matt
[root@serverX ~]# passwd matt
Changing password for user matt.
New password: matt
BAD PASSWORD: it is too short
BAD PASSWORD: is too simple
Retype new password: matt
passwd: all authentication tokens updated successfully.
```

11. Create a user account named **cindy** using a password of **cindy**.

```
[root@serverX ~]# useradd cindy
[root@serverX ~]# echo cindy | passwd --stdin cindy
Changing password for user cindy.
passwd: all authentication tokens updated successfully.
```

12. Create a group named **admins** that includes **matt** and **cindy**.

```
[root@serverX ~]# groupadd -r admins
[root@serverX ~]# usermod -aG admins matt
[root@serverX ~]# usermod -aG admins cindy
```

13. Configure Samba to share the **/test/** directory using a share name of **test**. Make it readable for **cindy** (use a Samba password of **password**) and writable for **matt** (use a

Samba password of **password**). Make sure the Linux permissions allow read/write as listed here, as well as meeting the user, group and permission requirements listed above.

```
[root@serverX ~]# yum -y install samba
```

Add the following to the bottom of the `/etc/samba/smb.conf` file:

```
[test]
path = /test
writable = no
write list = matt
```

```
[root@serverX ~]# smbpasswd -a matt
New SMB password: password
Retype new SMB password: password
[root@serverX ~]# smbpasswd -a cindy
New SMB password: password
Retype new SMB password: password
[root@serverX ~]# service smb start
[root@serverX ~]# chkconfig smb on
```

Change the `/test` line in `/etc/fstab` so that it includes the **acl** option, and make ACL option available to the mounted filesystem.

```
LABEL=test      /test    ext4      _netdev,acl 1 2
```

```
[root@serverX ~]# mount -o remount /test
[root@serverX ~]# setfacl -m u:matt:rwX /test
[root@serverX ~]# setfacl -m u:cindy:rx /test
[root@serverX ~]# semanage fcontext -a -t public_content_rw_t '/test(/.*)?'
[root@serverX ~]# restorecon -RFvv /test
[root@serverX ~]# setsebool -P allow_smbd_anon_write=1
[root@serverX ~]# iptables -I INPUT 3 -m state --state NEW -s 192.168.0.0/24 -p tcp --dport 445 -j ACCEPT
[root@serverX ~]# service iptables save
```

14. Configure a secure web server using the certificate and key located at `http://instructor/pub/materials/tls/certs/serverX.crt` and `http://instructor/pub/materials/tls/private/serverX.key`. Make the web server use `/mylv/index.html` as the default web page. Configure the `index.html` file such that accessing the secure web site will present the following:

```
Hello World!
```

```
[root@serverX ~]# yum install -y mod_ssl
[root@serverX ~]# wget http://instructor/pub/materials/tls/private/serverX.key -O /etc/pki/tls/private/serverX.key
[root@serverX ~]# chmod 600 /etc/pki/tls/private/serverX.key
[root@serverX ~]# wget http://instructor/pub/materials/tls/certs/serverX.crt -O /etc/pki/tls/certs/serverX.crt
```

Change the certificate file locations in `/etc/httpd/conf.d/ssl.conf` that currently point to localhost:

```
SSLCertificateFile /etc/pki/tls/certs/serverX.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/serverX.key
```

Edit **/etc/httpd/conf/httpd.conf** to change the default location:

```
DocumentRoot "/mylv"
```

```
[root@serverX ~]# service httpd start
[root@serverX ~]# chkconfig httpd on
[root@serverX ~]# iptables -I INPUT 3 -m state --state NEW -s 192.168.0.0/24 -p tcp --
dport 443 -j ACCEPT
[root@serverX ~]# iptables -I INPUT 3 -m state --state NEW -s 192.168.0.0/24 -p tcp --
dport 80 -j ACCEPT
[root@serverX ~]# service iptables save
[root@serverX ~]# semanage fcontext -a -t httpd_sys_content_t '/mylv(/.*)?'
[root@serverX ~]# restorecon -RFvv /mylv/
```

15. Allow **cindy** and **matt** to write the **/mylv/index.html** file.

```
[root@serverX ~]# chgrp admins /mylv/index.html
[root@serverX ~]# chmod 664 /mylv/index.html
```

Extra Review Checklist

- ☐ Create a new 100MB partition with an ext4 filesystem that can be mounted on **/encrypted**. Encrypt the filesystem using a password of **encrypted**. Place an entry in **/etc/fstab**, but do not allow it to automatically mount at boot.
- ☐ Extend the **mylv** logical volume and its filesystem to 2GB.
- ☐ Configure a VNC server for **student** with a VNC password of **test123**.
- ☐ Configure a logging server such that anyone in the local subnet can send logs to **TCP/514**.
- ☐ Create an RPM package named **test-1.0-1.el6.noarch.rpm**. Include the **/root/bin/test.sh** file in the RPM package. The **test.sh** script should simply run the **ls** command.
- ☐ Configure full sudo access for **matt** and **cindy**.

- ☐ Configure a caching-only name server.