



# Red Hat

RH300

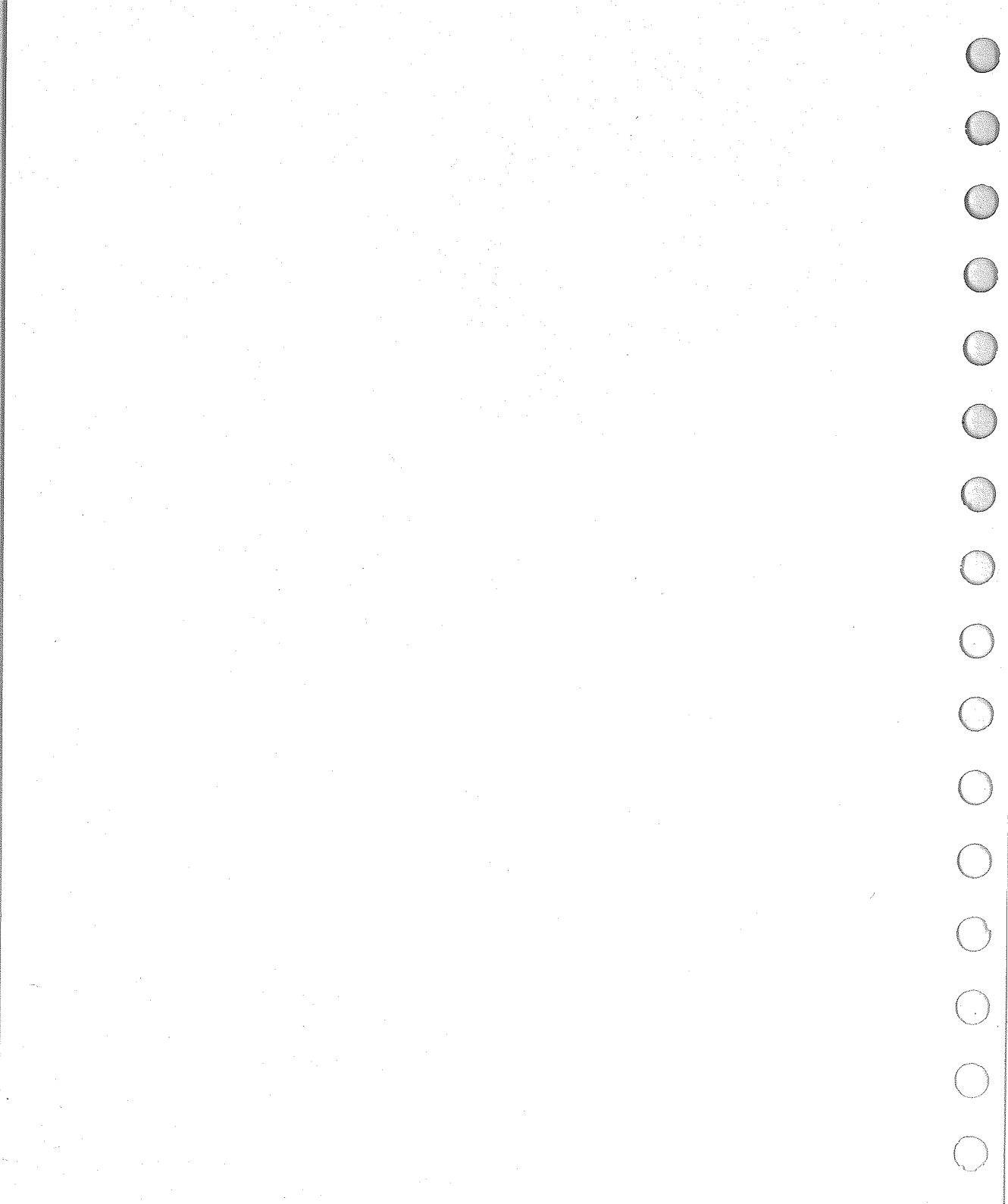
RHCE "Rapid Track" Course

RH300-RHEL5-en-2-EMEA-20070602

Red Hat Europe, 10 Alan Turing Road,  
Guildford, Surrey. GU2 7YF.  
United Kingdom

Tel: +(44)-1483-300169

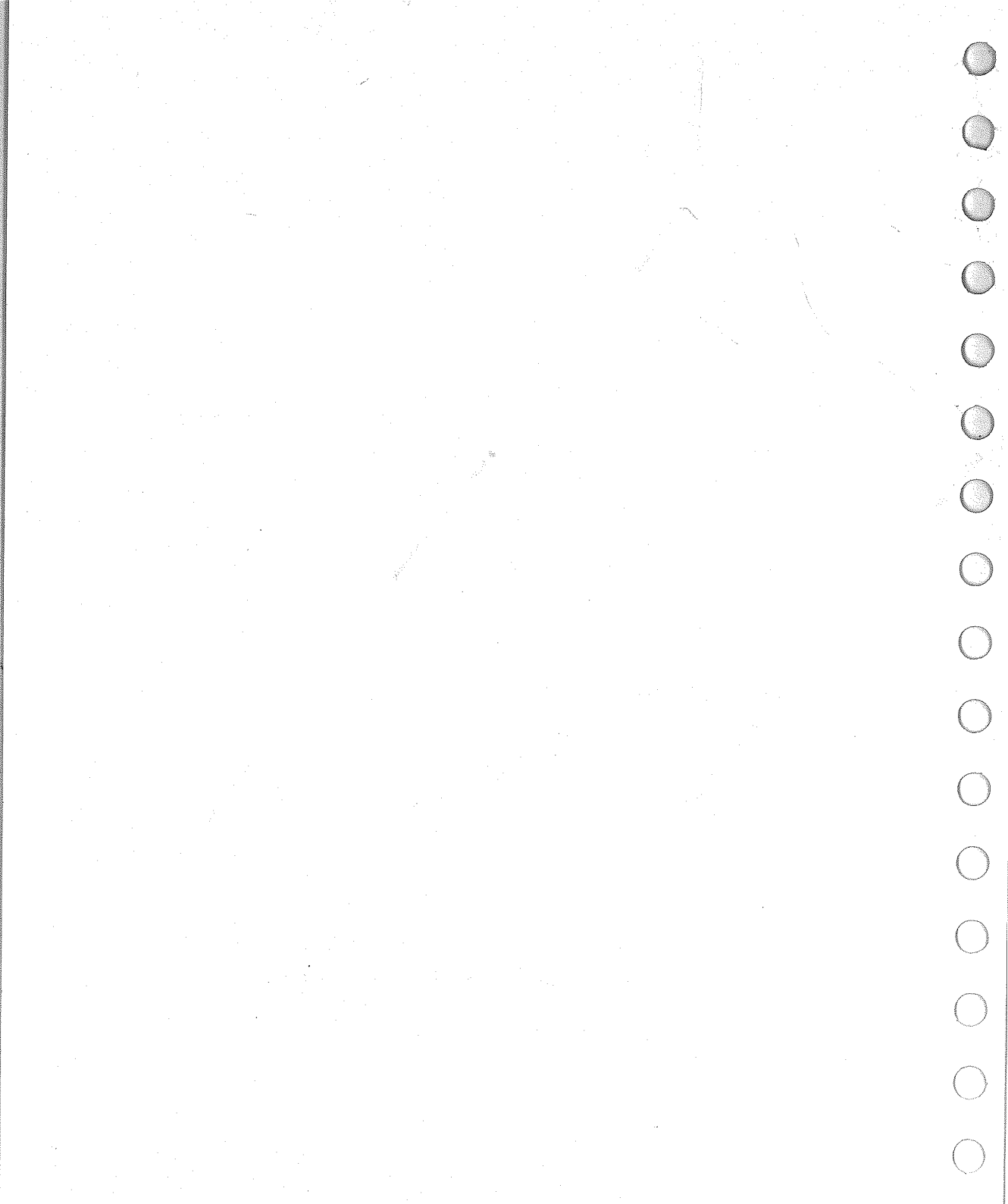
FAX: +(44)-1483-574944



# RH300

RHCE “Rapid Track” Course

RH300-RHEL5-en-2-EMEA-20070602





# Table of Contents

## RH300 - RHCE "Rapid Track" Course

### RHCE "Rapid Track" Course

Copyright	x
Welcome	xi
Participant Introductions	xii
Red Hat Enterprise Linux	xiii
Red Hat Enterprise Linux Variants	xiv
Red Hat Network	xv
Other Red Hat Supported Software	xvi
The Fedora Project	xvii
Classroom Network	xviii
Notes on Internationalization	xix
Objectives of RH300	xx
Audience and Prerequisites	xxi
The Big Picture	xxii

### Unit 1 - Essentials

Objectives	2
Virtual Consoles	3
Editors	4
Switching Accounts	5
Adding a New User Account	6
Using <b>cron</b>	7
Managing Ethernet Connections	8
Non-GUI Mail Clients	9
The OpenSSH Client	10
End of Unit 1	12
<b>Lab 1: Essentials</b>	<b>13</b>
Sequence 1: Accessing and customizing the system	14

### Unit 2 - Kernel Services

Objectives	19
Kernel Images and Variants	20
Kernel Modules	21
Kernel Module Utilities	22
Kernel Configuration With <code>/proc</code>	24
<code>/proc</code> Examples	25
<b>sysctl</b> : Persistent Kernel Configuration	26
Accessing Drivers Through <code>/dev</code>	27
Device Node Examples	28
Managing <code>/dev</code> With <code>udev</code>	29

Adding Files Under /dev	30
Exploring Hardware Devices	31
End of Unit 2	32
<b>Lab 2: Configuring the kernel</b>	<b>33</b>
Sequence 1: Turning off ping responses	34
Optional Sequence 2: Creating a file persistently under /dev/	35

## Unit 3 - Filesystem Management

Objectives	39
<b>fdisk</b>	<b>40</b>
Making Filesystems	42
Filesystem Labels	44
Mount Points and /etc/fstab	45
Unmounting Filesystems	47
Handling Swap Files and Partitions	48
End of Unit 3	49
<b>Lab 3: Creating Filesystems</b>	<b>50</b>
Sequence 1: Create a new filesystem	51
Sequence 2: Creating a new swap partition	52

## Unit 4 - User Administration

Objectives	59
Modifying User Accounts	60
Group Administration	62
Password Aging Policies	63
Deleting Accounts	64
SGID Directories	65
The Sticky Bit	67
Configuring the Quota System	68
Setting Quotas for Users	69
Reporting Quota Status	71
End of Unit 4	72
<b>Lab 4: User and Group Administration</b>	<b>73</b>
Sequence 1: Creating the groups and users	74
Sequence 2: Setting up shared directories	75
Sequence 3: Implementing Quotas	76

## Unit 5 - Local Security

Objectives	84
Default Firewall	85
Access Control List (ACL)	86
ACL Usage	87
ACL Inheritance	88
SELinux	89
SELinux, continued	90

SELinux: Targeted Policy	92
SELinux: Management	94
SUID and SGID Executables	96
sudo	97
System Logging	98
End of Unit 5	99
<b>Lab 5: Local Security</b>	<b>100</b>
Sequence 1: Working with ACLs	101
Sequence 2: Understanding file context.	102
Sequence 3: SELinux Booleans	103

## Unit 6 - Advanced Partitioning

Objectives	112
What is Software RAID?	113
Software RAID Configuration	114
Software RAID Testing and Recovery	115
What is Logical Volume Manager (LVM)?	116
Creating Logical Volumes	117
Resizing Logical Volumes	119
End of Unit 6	121
<b>Lab 6: Advanced Partitioning</b>	<b>122</b>
Sequence 1: Working With Software RAID	123
Sequence 2: Creating A Logical Volume	124
Sequence 3: Extending A Logical Volume	125
Sequence 4: Reduce a Logical Volume	126

## Unit 7 - Installation

Objectives	135
Anaconda, the Red Hat Enterprise Linux Installer	136
First Stage: Starting the Installation	137
First Stage: Boot Media	138
Accessing the Installer	140
First Stage: Installation Method	141
Second Stage: Installation Overview	142
Configuring File Systems	143
Advanced Partitioning	144
Package Selection	145
First Boot: Post-Install Configuration	146
Kickstart	147
Starting a Kickstart Installation	148
Anatomy of a Kickstart File	150
Kickstart: Commands Section	151
Kickstart: Commands section	152
Kickstart: Packages Section	154
Kickstart: %pre, %post	155
End of Unit 7	156

<b>Lab 7: Installation and System-Initialization</b>	<b>157</b>
Sequence 1: Installing Red Hat Enterprise Linux	158
Sequence 2: Kickstart Installation	160

## Unit 8 - System Initialization

Objectives	168
Boot Sequence Overview	169
BIOS Initialization	170
Bootloader Components	171
GRUB and <code>grub.conf</code>	172
Starting the Boot Process: GRUB	174
The Chicken/Egg Module Problem and the Initial RAM Disk	175
Kernel Initialization	177
init Initialization	178
Run Levels	179
<code>/etc/rc.d/rc.sysinit</code>	180
<code>/etc/rc.d/rc</code>	181
System V run levels	182
<code>/etc/rc.d/rc.local</code>	183
Controlling Services	184
End of Unit 8	185
<b>Lab 8: Managing Startup</b>	<b>186</b>
Sequence 1: Changing the default run level	187
Sequence 2: Exploring an initial RAM disk	188
Sequence 3: GRUB	190

## Unit 9 - RPM, YUM, RHN

Objectives	196
RPM Package Manager	197
Installing and Removing Software	198
Updating a Kernel RPM	200
rpm Queries	201
rpm Verification	202
About yum	203
Using yum	205
Searching packages/files	206
Configuring Additional Repositories	207
Red Hat Network	208
Red Hat Network Server	209
Entitlements	210
Red Hat Network Client	211
End of Unit 9	212
<b>Lab 9: Working with packages</b>	<b>213</b>
Sequence 1: Using RPM	214
Sequence 2: Connecting to a private repository	215
Sequence 3: Installing new packages using yum	216

## Unit 10 - System Administration Topics

Objectives	224
XOrg Server Configuration	225
CUPS	226
System crontab Files	227
Daily Cron Jobs	228
The <b>anacron</b> System	230
Automounter	231
PAM Operation	233
/etc/pam.d/ Files: Tests	234
/etc/pam.d/ Files: Control Values	235
Important PAM Modules	236
End of Unit 10	237
<b>Lab 10: System Administration</b>	<b>238</b>
Sequence 1: CUPS printer administration.	239

## Unit 11 - Network Configuration

Objectives	243
Network Configuration Files	244
Network Configuration Tools	245
Address Types	246
Address Types - part 2	247
Address Representation	248
New and Modified Utilities	249
OpenSSH Overview	250
OpenSSH Server Configuration	251
VNC: Virtual Network Computing	253
Authentication Configuration	254
Example: NIS Configuration	256
Example: LDAP Configuration	257
The <b>xinetd</b> service	259
<b>xinetd</b> service controls	260
Network Diagnostic Tools	261
End of Unit 11	262
<b>Lab 11: Network</b>	<b>263</b>
Sequence 1: Using IPv6	264
Sequence 2: Exploring Xinetd Services	265
Sequence 3: Client-side NIS account management	266
Sequence 4: Client-side LDAP account management	267

## Unit 12 - Network Security

Objectives	277
tcp_wrappers Configuration	278

Daemon Specification	279
Client Specification	280
Advanced Client Syntax	281
tcp_wrappers Example	282
Netfilter Packet Flow	283
Rule Matching	284
Rule Targets	285
Simple Example	286
Basic Chain Operations	287
Additional Chain Operations	288
Common Match Criteria	289
Common Match Criteria	290
Rules Persistence	291
End of Unit 12	292
<b>Lab 12: Network Security</b>	<b>293</b>
Sequence 1: Restricting services with tcp_wrappers	294
Sequence 2: Applying simple packet filtering to a host	295

## Unit 13 - Network File Sharing Services

Objectives	302
File Transfer Protocol (FTP)	303
FTP Security	304
FTP Configuration	306
Network File Service (NFS)	307
NFS Security	308
NFS Optional Firewall Ports	309
NFS Configuration	310
NFS Client-side	311
Samba (SMB)	312
SMB Security	313
SMB Configuration	314
SMB Configuration, cont	315
SMB Passwords	316
SMB Client-side	317
End of Unit 13	319
<b>Lab 13: Network File Sharing Services</b>	<b>320</b>
Sequence 1: Implementing FTP Services	321
Sequence 2: Implementing NFS Services	322
Sequence 3: Implementing SMB Services	323

## Unit 14 - Network Infrastructure

Objectives	336
Enabling Network Logging	337
Network Installation Server	338
Creating a private repository	339
Configuring an IPv4 DHCP Server	340

Basic Design of NTP	342
Server Configuration	343
DNS Overview	344
Berkeley Internet Name Domain	346
BIND: <code>named.conf</code>	347
BIND: Zone Files	349
Securing Infrastructure Services	350
End of Unit 14	351
<b>Lab 14: Enterprise Infrastructure</b>	<b>352</b>
Sequence 1: Logging to a centralized log host	353
Sequence 2: Working With BIND	354
Sequence 3: Migrating to a Zone Server	355

## Unit 15 - HTTP Service

Objectives	365
Apache Overview	366
Apache Security	367
Apache Server Configuration	368
Creating an Alternate DocumentRoot	369
Virtual Host Example	371
Apache httpd Access Control Example	373
Squid Web Proxy Cache	374
Useful parameters in <code>/etc/squid/squid.conf</code>	375
End of Unit 15	376
<b>Lab 15: HTTP Services</b>	<b>377</b>
Sequence 1: Apache installation and configuration	378
Sequence 2: Migrating to a Virtual Web server	379
Sequence 3: Basic Squid configuration	380

## Unit 16 - Mail Service

Objectives	389
An Email Review	390
Simple Mail Transport Protocol	391
Using <b>alternatives</b> to Switch MTAs	392
Mail Security	393
Sendmail Configuration Files	395
Incoming Sendmail Configuration	396
Sendmail Operation	397
Incoming Postfix Configuration	398
Postfix Operation	399
Email Aliases	400
Mail Retrieval Protocols	401
Dovecot Configuration	402
Verifying IMAP Operation	404
End of Unit 16	405
<b>Lab 16: Mail Services</b>	<b>406</b>

Sequence 1: Configure Sendmail as an MTA	407
Sequence 2: Migrating to Postfix	408
Sequence 3: Adding new aliases	409
Sequence 4: Installing the Dovecot MDA.	410
Sequence 5: Creating a unique Dovecot certificate	411

## Unit 17 - Troubleshooting

Objectives	424
Method of Fault Analysis	425
Fault Analysis: Gathering Data	426
Things to Check: X	428
Things to Check: Networking	429
Order of the Boot Process	430
Filesystem Problems During Boot	431
Recovery Run-levels	432
Rescue Environment	433
Rescue Environment Utilities	434
Rescue Environment Details	435
End of Unit 17	436
<b>Lab 17: System Rescue and Troubleshooting</b>	<b>437</b>
Sequence 1: Repairing the MBR in the rescue environment	438
Sequence 2: Installing software in rescue mode	439
Sequence 3: Troubleshooting Practice	440



# Introduction

## RHCE "Rapid Track" Course

1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

# Copyright

- The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2007 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 email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 866 626 2994 or +1 919 754 3700.

2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Welcome

Please let us know if you have any special needs while at our training facility.

3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Phone and network availability

Please only make calls during breaks. Your instructor will show you which phone to use. Network access and analog phone lines may be available; your instructor will provide information about these facilities. Please turn pagers to silent and cell phones off during class.

## Restrooms

Your instructor will notify you of the location of these facilities.

## Lunch and breaks

Your instructor will notify you of the areas to which you have access for lunch and for breaks

## In case of Emergency

Please let us know if anything comes up that will prevent you from attending.

## Access

Each facility has its own opening and closing times. Your instructor will provide you with this information.

# Participant Introductions

Please introduce yourself to the rest of the class!

4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Red Hat Enterprise Linux

- Enterprise-targeted operating system
- Focused on mature open source technology
- 18-24 month release cycle
  - Certified with leading OEM and ISV products
- Purchased with one year Red Hat Network subscription and support contract
  - Support available for seven years after release
  - Up to 24x7 coverage plans available

5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Red Hat Enterprise Linux product family is designed specifically for organizations planning to use Linux in production settings. All products in the Red Hat Enterprise Linux family are built on the same software foundation, and maintain the highest level of ABI/API compatibility across releases and errata. Extensive support services are available: a one year support contract and Update Module entitlement to Red Hat Network are included with purchase. Various Service Level Agreements are available that may provide up to 24x7 coverage with guaranteed one hour response time. Support will be available for up to seven years after a particular release.

Red Hat Enterprise Linux is released on an eighteen to twenty-four month cycle. It is based on code developed by the open source community and adds performance enhancements, intensive testing, and certification on products produced by top independent software and hardware vendors such as Dell, IBM, Fujitsu, BEA, and Oracle. Red Hat Enterprise Linux provides a high degree of standardization through its support for five processor architectures five processor architectures (Intel x86-compatible, AMD AMD64/Intel 64, Intel Itanium 2, IBM POWER, and IBM mainframe on System z).

# Red Hat Enterprise Linux Variants

- Two Install Sets available
- Server
  - Red Hat Enterprise Linux Server
  - Red Hat Enterprise Linux Advanced Platform
- Desktop
  - Red Hat Enterprise Linux Desktop
  - Workstation Option
  - Multi-OS Option

6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Currently, on the x86-compatible architecture, the product family includes:

*Red Hat Enterprise Linux Advanced Platform:* the most cost-effective server solution, this product includes support for the largest x86-compatible servers, unlimited virtualized guest operating systems, storage virtualization, high-availability application and guest fail-over clusters, and the highest levels of technical support.

*Red Hat Enterprise Linux:* the basic server solution, supporting servers with up to two CPU sockets and up to four virtualized guest operating systems.

*Red Hat Enterprise Linux Desktop:* a general purpose client solution, offering desktop applications such as the OpenOffice.org office suite and Evolution mail client. Add-on options provide support for high-end development workstations and virtualization.

# Red Hat Network

- A comprehensive software delivery, system management, and monitoring framework
  - *Update Module*: Provides software updates
    - Included with all Red Hat Enterprise Linux subscriptions
  - *Management Module*: Extended capabilities for large deployments
  - *Provisioning Module*: Bare-metal installation, configuration management, and multi-state configuration rollback capabilities
  - *Monitoring Module* provides infrastructure health monitoring of networks, systems, applications, etc.

7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Network is a complete systems management platform. It is a framework of modules for easy software updates, systems management, and monitoring, built on open standards. There are currently four modules in Red Hat Network; the Update Module, the Management Module, the Provisioning Module, and the Monitoring Module.

The Update Module is included with all subscriptions to Red Hat Enterprise Linux. It allows for easy software updates to all your Red Hat Enterprise Linux systems.

The Management Module is an enhanced version of the Update Module, which adds additional features tailored for large organizations. These enhancements include system grouping and set management, multiple organizational administrators, and package profile comparison among others. In addition, with RHN Proxy Server or Satellite Server, local package caching and management capabilities become available.

The Provisioning Module provides mechanisms to provision and manage the configuration of Red Hat Enterprise Linux systems throughout their entire life cycle. It supports bare metal and existing state provisioning, storage and editing of kickstart files in RHN, configuration file management and deployment, multi-state rollback and snapshot-based recovery, and RPM-based application provisioning. If used with RHN Satellite Server, support is added for PXE bare-metal provisioning, an integrated network tree, and configuration management profiles.

## Other Red Hat Supported Software

- Global Filesystem
- Directory Server
- Certificate Server
- Red Hat Application Stack
- JBoss Middleware Application Suite

8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat offers a number of additional open source application products and operating system enhancements which may be added to the standard Red Hat Enterprise Linux operating system. As with Red Hat Enterprise Linux, Red Hat provides a range of maintenance and support services for these add-on products. Installation media and software updates are provided through the same Red Hat Network interface used to manage Red Hat Enterprise Linux systems.

*Global Filesystem:* an open source cluster file system appropriate for enterprise deployments, allowing servers to share a common pool of storage.

*Directory Server:* an LDAP-based server that centralizes directory storage and data distribution, such as user and group data.

*Certificate Server:* identity management software, using the Red Hat Directory Server as its back-end LDAP data repository.

*Red Hat Application Stack:* the first fully integrated open source stack, supplying everything needed to run standards based web applications, including Red Hat Enterprise Linux, JBoss Application Server with Tomcat, JBoss Hibernate, and a choice of open source databases: MySQL or PostgreSQL, and Apache Web Server.

*JBoss Middleware Application Suite:* a suite of applications that provide a complete middleware solution.

For additional information, see the following web pages:

- Global Filesystem: <https://www.redhat.com/solutions/gfs/>
- Directory Server: <https://www.redhat.com/solutions/directoryserver/>
- Red Hat Application Stack:  
<https://www.redhat.com/solutions/rhappstack/>
- JBoss Middleware Application Suite: <https://www.redhat.com/jboss/>



# The Fedora Project

- Red Hat sponsored open source project
- Focused on latest open source technology
  - Rapid four to six month release cycle
  - Available as free download from the Internet
- An open, community-supported proving ground for technologies which may be used in upcoming enterprise products
- Red Hat does not provide formal support

9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Fedora Project is a community supported open source project sponsored by Red Hat intended to provide a rapidly evolving, technology-driven Linux Distribution with an open, highly scalable development and distribution model. It is designed to be an incubator and test bed for new technologies that may be used in later Red Hat enterprise products. The basic Fedora Core distribution will be available for free download from the Internet.

The Fedora Project will produce releases on a short four to six month release cycle, to bring the latest innovations of open source technology to the community. This may make it attractive for power users and developers who want access to cutting-edge technology and can handle the risks of adopting rapidly changing new technology. Red Hat does not provide formal support for the Fedora Project.

For more information, visit <http://www.fedoraproject.org>.

## Classroom Network

	Names	IP Addresses
Our Network	example.com	192.168.0.0/24
Our Server	server1.example.com	192.168.0.254
Our Stations	stationx.example.com	192.168.0.x
Hostile Network	cracker.org	192.168.1.0/24
Hostile Server	server1.cracker.org	192.168.1.254
Hostile Stations	stationx.cracker.org	192.168.1.x
Trusted Station	trusted.cracker.org	192.168.1.21

10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Notes on Internationalization

- Red Hat Enterprise Linux supports nineteen languages
- Default language can be selected:
  - During installation
  - With **system-config-language**
    - System->Administration->Language
- Alternate languages can be used on a per-command basis:  

```
$ LANG=en_US.UTF8 date
```
- Language settings are stored in `/etc/sysconfig/i18n`

11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux 5 supports nineteen languages: English, Bengali, Chinese (Simplified), Chinese (Traditional), French, German, Gujarati, Hindi, Italian, Japanese, Korean, Malayalam, Marathi, Oriya, Portuguese (Brazilian), Punjabi, Russian, Spanish and Tamil. Support for Assamese, Kannada, Sinhalese and Telugu are provided as technology previews.

A system's language can be selected during installation, but the default is US English. To use other languages, you may need to install extra packages to provide the appropriate fonts, translations and so forth. These can be selected during system installation or with **system-config-packages** (Applications->Add/Remove Software).

The currently selected language is set with the `LANG` shell variable. Programs read this variable to determine what language to use for output:

```
[student@stationX ~]$ echo $LANG  
ru_RU.UTF8
```

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

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:

```
[student@stationX ~]$ LANG=en_US.UTF8 date  
Thu Feb 22 13:54:34 EST 2007
```

Subsequent commands will revert to using the system's default language for output.

SCIM (Smart Common Input Method) can be used to input text in various languages under X if the appropriate language support packages are installed. Type *Ctrl-Space* to switch input methods.

# Objectives of RH300

- **Technical Objective**
  - Train and certify Red Hat Enterprise Linux specific concepts and skills at the system administrator level
- **Readiness Objective**
  - Assure a minimum level of systems administration skills so that a Red Hat Certified Engineer, RHCE, is qualified for professional responsibilities in managing Red Hat Enterprise Linux systems

12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Technical skills definition of Red Hat Certified Engineer, RHCE, certificate:

A person who has passed the RHCE certification process has demonstrated basic skills required to administer Red Hat Enterprise Linux systems. These skills include:

- An understanding of Linux specific hardware issues
- The ability to install Red Hat Enterprise Linux
- An understanding of the X Window System and it's configuration
- The ability to configure basic networking, security, and to configure typical network services
- Demonstrate system administration, diagnostic, and troubleshooting skills

Readiness definition:

When demonstrated in a Certification Lab Exam based on realistic tasks, the above technical objectives assure a certain minimum level of technical readiness for professional duties in managing Red Hat Enterprise Linux systems.

Red Hat believes a lab-based certification is a better indication of an individual's readiness to administer Red Hat Enterprise Linux than a certification based on a multiple choice exam.

## Audience and Prerequisites

- Audience: Linux or UNIX system administrators, network specialists, and other UNIX or Linux power users
- Prerequisites: experience in UNIX or Linux at the power user, network operations, or system administrator level

13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Audience for RH300:

The Red Hat Certified Engineer course is designed for UNIX and Linux experienced users, networking specialists and system administrators who want to train and certify their skills on Red Hat Enterprise Linux at the level indicated.

Good candidates for RHCE consist of individuals who fit the following profiles:

- UNIX system administrators and UNIX system engineers
- UNIX or Linux network engineers, NOC and ISP technical staff
- Linux system administrators and engineers
- Other UNIX and/or Linux power users who may want focused training and certification on Red Hat Enterprise Linux

A full list of suggested exam candidate skills, along with other useful RHCE exam material, can be found on the RHCE Exam Preparatory Guide available at:

<http://www.redhat.com/training/rhce/examprep.html>

# The Big Picture

- Foundation: Building Blocks
  - System operations
- Management: Keep It Running
  - System level maintenance
- Networking: Connecting
  - Establishing and securing
- Services: The Enterprise
  - Infrastructure and applications

14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Red Hat Enterprise Linux RHCE "Rapid Track" Course material is broken into four broad categories. Please do not confuse these subject categories with a daily schedule, as some categories maybe covered in more detail than others.

System operations will cover material that an RHCE would expect to deal with on a daily basis. For a well prepared student, they will find much of this information to be review. For those transitioning from other similar operating systems, this maybe an opportunity to learn which commands support a given feature.

Systems level maintenance is far more Red Hat specific, and will be focused on much higher level administrative issues. As an example, an administrator should be prepared to add a user to a system on a minutes notice (operations), where as configuring storage arrays or preparing automated installs may require more preparation.

Establishing and securing network configuration will include local administration, as well as connection to a network on an individual system basis. This is the point where we will transition from being concerned about the box, to being concerned about the network.

Infrastructure and applications will focus on using the box as a server, and integrating into the enterprise. Not only will this include application servers that provide a single function, such as a web server, but also servers that provide the backbone services need by the network, as a whole.

A few notes of interest regarding the RH300, RHCE "Rapid Track" Course:

- This class is not a boot camp class. This is to say that it is not designed as a single, start to finish class, that providing the full spectrum of skills needed to fill the role of RHCE. There is a significant level of pre-requisite knowledge.
- This class is not an exam prep course. This is to say that we will focus on the skills needed to fill a position expecting RHCE level skills.

# Unit 1

## Essentials

1-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Use essential administrative tools
- Perform basic user configuration
- Configure basic networking

1-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Virtual Consoles

- Multiple non-GUI logins are possible through the use of virtual consoles
- There are by default 6 available virtual consoles
- Available through *Ctrl-Alt-F[1-6]*
- If X is running, it is available as *Ctrl-Alt-F7*

1-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *Virtual Consoles*

The virtual consoles enable a user to have multiple logins even when not using an X window system. They provide full screen, non-GUI access to the system. The virtual console available through *Ctrl-Alt-F7* runs the X window system when X is running.

You can scroll at the virtual consoles by using *Shift-PgUp* and *Shift-PgDn*. Be aware that the scroll buffer is stored in video memory, so if you change virtual consoles, the scroll buffer will be lost.

# Editors

- **vi (vim)** - the default editor
  - To edit a file: **vi** /path/filename
  - Press *Insert* to make changes.
  - Press *Escape* when finished editing.
  - To save and exit: **:wq**
- **nano**
  - To edit a file: **nano** /path/filename
  - To save and exit: **Ctrl-O** then **Ctrl-x**
  - Modify /etc/bashrc:

```
EDITOR=/usr/bin/nano; export EDITOR
```

1-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux contains many plain text editors. Arguably, the most popular, most powerful, and most ubiquitous plain text editor in the Linux and UNIX worlds is **vi** and the upgraded version, **vim**. As the default editor, **vim** is called when ever a text terminal needs to interactively modify a file. For all its powers and features, there are really very few that are needed for day to day operation.

At the very least, you should know that to add content, you will have to press *Insert* to change from the *Command* mode to the *Insert* mode. Once the changes have been made, press *Escape* to return to the *Command* mode. To save the changes, Press **:q-w** followed by *Enter*.

One of the easiest editors to learn is **nano**, a plain text editor that runs in a terminal window. You may specify a file on the command line by giving the filename as an argument. If the file exists, you will edit it; if it does not exist, **nano** will create the file when you save it.

Once in **nano**, you simply type the text you wish to add; use the arrow keys to move the cursor around the file; or use the *Del* or *BkSpc* keys to forward delete or backspace over text.

Other commands in **nano** are run using the **Ctrl** key. The last two lines on the screen will display a menu of commands to run. Typically, the menu will look like this:

```
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Txt  ^T To Spell
```

This tells you, for example, that you can type **Ctrl-x** to exit (ignore the case difference; in this instance, it does not matter).

Regardless of your choice, it is important to understand that **vim** is the default. To change the default, you will have to set the **editor** variable, and **export** the value. This change will have to be committed to a startup file, in order for the change to be persistent across reboots.

# Switching Accounts

- Syntax
  - **su [-] [user]**
  - **su [-] [user] -c *command***
- Allows the user to temporarily become another user
  - Default user is root
- The “-” option makes the new shell a login shell

1-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2984 or +1 (919) 754 3700.

The **su** command is used to switch to another account from the command line. This command is most often used by system administrators to temporarily become the root user without logging out of their non-privileged account. It is very important to only use the root account when absolutely necessary, because of the awesome power of the root account. Day to day use of the system should never be conducted with the root account.

The password of the account being switched to must be supplied unless the superuser issued the **su** command.

Without the - option, the original user's environment is maintained. Using the - option causes the new shell to be a login shell which, among other things, unsets the original user's environment variables in the new shell. This is usually preferred so that actions performed as the new user will not have any effect on the old user, and so the new user's environment is available, rather than the old.

There are a number of options to the **su** command. See the documentation for more information.

Most systems log the use (or attempted use) of **su** to change to the root account. System administrators will often log on to the system as an ordinary user and then use **su** to gain access to the root account rather than log in directly as root so that the switch is logged.

Most systems administration tasks are best performed using **sudo**. **sudo** is safer than an **su** to root.

# Adding a New User Account

- Most common method is **useradd**:
  - **useradd** [*options*] *username*
- Running **useradd** is equivalent to:
  - editing `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`
  - creating and populating home directory
  - setting permissions and ownership
- Set account password using **passwd**

1-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command-line utility `useradd` provides a simple method for adding new users to the system:

```
[root@stationX] # useradd joshua
```

The above command will add a new account to the machine called `joshua` as well as set up that user's home directory, and create a private group for the user, also called `joshua`. The next step would be to assign `joshua` a password which you can do by simply typing the following command:

```
[root@stationX] # passwd joshua
```

# Using cron

- Edit your **cron** file
  - **crontab** [-l|-r|-e]
- Root can modify users **crontab**'s
  - **crontab** [-u user] [-l|-r|-e]
- Entry consists of five space-delimited fields followed by a command line
- Fields are minute, hour, day of month, month, and day of week
- Comment lines begin with #
  - 55 9 25 12 \* echo "Dec 25th"
  - 55 9 \* \* 1 echo "Monday"

1-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *Using cron to schedule processes (continued)*

Cron files ("crontabs") are stored in `/var/spool/cron`, which is not accessible by non-privileged users. In order to access the current crontab, the **crontab** command is used.

To install or modify a cron table, use **crontab**'s **-e** option. The **-l** option displays the current crontab file and the **-r** option removes it.

## *Crontab examples*

Entry fields can be separated by any number of tabs or spaces. Valid field values are as follows:

Minute#0-59

Hour#0-23

Day of Month#1-31

Month#1-12

Day of Week#0-6 (0 = Sunday)

Multiple values may be separated by commas. An asterisk in a field represents all valid values. A user's crontab may look like the following:

#Min	Hour	DoM	Month	DoW	Command
0	4	*	*	1,3,5	find ~ -name core   xargs rm -f {}
0	0	31	10	*	mail -s "boo" \$LOGNAME < boo.txt

# Managing Ethernet Connections

- Network interfaces are named sequentially: `eth0`, `eth1`, etc
  - Multiple addresses can be assigned to a device with *aliases*
  - Aliases are labeled `eth0:1`, `eth0:2`, etc.
  - Aliases are treated like separate interfaces
- View interface configuration with **`ifconfig [ethx]`**
- Enable interface with **`ifup ethx`**
- Disable interface with **`ifdown ethx`**

1-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Network connection names consist of a prefix, based on the device type, and a number to distinguish a particular device from others of its type. For example, all ethernet devices have the prefix `eth`. The first detected ethernet card is assigned the name `eth0`, the second `eth1` and so forth. Every system also has a special network device called the `lo`, which represents the "localhost" or "loopback" device with address `127.0.0.1`. You can view the basic settings of a network device by running the **`ifconfig`** command. By default, **`ifconfig`** will print information on all active devices. If given a device name as an argument, it will print information about that device only:

```
[student@stationX ~]$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:09:6B:CD:2B:87
          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::209:6bff:fecd:2b87/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:851525 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1132322 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:211140434 (201.3 MiB)  TX bytes:1113058956 (1.0 GiB)
```

While at first glance this output may be a bit overwhelming, it is helpful to know that the most important information is in the top three lines. The first line tells us that `eth0` is an ethernet device with the hardware (or MAC) address `00:09:6B:CD:2B:87`. This is a unique address built into every ethernet card by the manufacturer and can be useful for identifying specific devices on the network.

The second line lists the three fundamental IP configuration parameters: the IP address (`192.168.0.254`), broadcast address (`192.168.0.255`) and network mask (`255.255.255.0`). These settings are used to identify your system to other machines on the network and define how your system interacts with others. A detailed explanation of how each setting works is beyond the scope of this class, but we will be seeing how each can be set, both automatically and manually, in subsequent slides.

An interface can be configured but not running. Such an interface is said to be *disabled* or *down*. If an interface is down it will not be shown in **`ifconfig`**'s output unless the name of the device is passed as an argument. Devices can be brought up and down by an administrator using the **`ifup`** and **`ifdown`** commands.

# Non-GUI Mail Clients

- **mutt**
  - Supports pop, imap and local mailboxes
  - Highly configurable
  - Mappable hotkeys
  - Message threading and colorizing
  - GnuPG integration
  - Context-sensitive help with '?'

1-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *Introducing Mutt*

It is important to know how to access email even when a graphical interface is not available. The **mutt** email client is a powerful tool for managing email within a text-only environment. Many Linux users even prefer it over evolution. Some of the strengths of **mutt** are its handling of message threads and its flexibility. Every hotkey can be mapped to your liking and whole strings of actions can be bound to a single keystroke. When **mutt** sees a series of related messages (an original message, followed by a series of replies on a mailing list, for example), it displays each reply under the parent. These *threads* can then be navigated, deleted and marked as read or unread as a group instead of one at a time. This makes managing high volumes of email much easier.

## *Mailboxes in mutt*

You read one "mailbox" (a local mail spool, imap account or pop account) at a time when using **mutt**. You can specify the mailbox you wish to start in by running mutt with the **-f** argument. For example:

```
[student@stationX ~]$ mutt -f imaps://user@server
```

If no mailbox is specified then your local mail spool will be viewed. While in mutt you can switch mailboxes by pressing the **c** key and typing the url of the mailbox you would like to read. You can change which mailbox mutt views by default by altering its configuration file, `~/.muttrc`.

## *Documentation and Help*

Documentation and example `.muttrc` files are available in mutt's `/usr/share/doc/` subdirectory. mutt has a large number of very powerful key-bindings for managing large amounts of email. Fortunately it also has context-sensitive help to assist you in finding out how to perform a given task. For example, if you press the **?** key while viewing the message-list for a mailbox, the key-bindings for managing a mailbox will be displayed. If you press **?** while viewing a message then the key-bindings for navigating a message will be displayed. The most commonly used commands will always be displayed at the top of your screen.

# The OpenSSH Client

- Secure shell sessions
  - `ssh hostname`
  - `ssh user@hostname`
  - `ssh hostname remote-command`
- Secure remote copy files and directories
  - `scp file user@host:remote-dir`
  - `scp -r user@host:remote-dir localdir`
- Secure ftp provided by `sshd`
  - `sftp host`
  - `sftp -C user@host`

1-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Several common, and notoriously insecure applications, such as **telnet**, **rsh**, **rcp**, and **ftp** may be replaced with **ssh**, **scp**, and **sftp**. These include **telnet**, **rsh**, **rcp**. While well known, and at times, useful, these applications pass both authentication information and data as "clear text", or unencrypted packets. Such data can easily be viewed by a third party using standard tools such as **tcpdump** and **wireshark**. In contrast, both initial authentication and subsequent data transfer are encrypted when using **ssh**.

Another advantage of SSH over the "r commands" lies in its mechanisms for transparent authentication. Instead of simply trusting connections from IP addresses listed in `$HOME/.rhosts` or `/etc/hosts.equiv`, **ssh** makes use of cryptographic keys to strongly authenticate incoming connections. These keys are stored in `$HOME/.ssh/authorized_keys` or `/etc/ssh/ssh_known_hosts`.

**ssh** is fairly simple to use. By default, **ssh** uses the local username to connect, but this can be overridden with `user@host` or `-l user`. When an **ssh** connection is made for the first time to a remote system, an entry will be appended to the local `~/.ssh/known_hosts` file that consists of the hostname plus the remote host's public **ssh** key. This key validates the host, in a sense: if the key changes, or if one has connected to a host that is not the original remote host, but now has that name, then **ssh** will exit with an error. **scp** uses a syntax much like that of **ssh**. It is possible to **scp** entire directories, and it is a more secure mechanism for transferring files than **ftp** or **rcp**. There are a few rules about the **-r** option that bear mention:

`scp -r localdir/ host:remote-dir` would copy the contents of `localdir` into

`remote-dir`, but there would not be a `remote-dir/dir` (a copy from the remote host would work the same.)

`scp -r localdir host:remote-dir` would copy both `localdir` and its contents to `remote-dir` on the remote host if it existed, but would behave like the previous example if the remote directory did not exist (a copy from the remote host would work the same).



**sftp -C *user@host*** would, using optional compression, log into *host* with a username of *user*, and present a password prompt and an interactive ftp-style session.

Note: system-wide default client configuration is set in `/etc/ssh/ssh_config`.

# End of Unit 1

- Questions and Answers
- Summary
  - su -, vi, nano
  - useradd, passwd
  - ifup, ifdown
  - mutt, ssh

1-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 1

## Essentials

---

Goal: Become familiar with basic system configuration and use.

System Setup: A system with a default configuration and a user named "student".

## Sequence 1: Accessing and customizing the system

**Scenario:** You have been provided a machine with a basic configuration that must be configured to provide access to additional users.

**Deliverable:** A system that is part of the classroom network.

### Instructions:

1. Logon to the GUI as "student" with the password of "student". Right click on the desktop and open a terminal. Change the password for "student" to a password you will remember.
2. Build a **cron** job that will execute a listing of student's home directory in ten minutes from now.
3. Launch the Firefox web browser by clicking the globe icon on the menu bar at the top of the screen. Complete the web page at <http://server1/cgi-bin/roster.cgi>.
4. Open a second terminal, and SSH to localhost as "visitor" with the password of "password". Do not change this user's password. Exit the SSH session.
5. In one of your terminal windows, use the **su** to become "root" with the password of "redhat". Use the **ifconfig** command to determine your IP address.
6. Add two of your classmates as users on your system. Give them the password of "unpriv".
7. Have the two new users SSH to your station. Login to two other stations on the network.
8. Use **ftp** to connect to `server1` using the user name *anonymous*. No password is needed. In the `pub` directory, download `getme`.
9. Use **vi** to add your name as the first line of this file.
10. Use SCP to send the edited file to the systems where you have a login account.
11. As root, verify that you have received a file from the users with an account on your system.
12. As student, use **mutt** to view the output of your cron job.

## Sequence 1 Solutions

1. Logon to the GUI as "student" with the password of "student". Right click on the desktop and open a terminal. Change the password for "student" to a password you will remember.

```
$ passwd
Changing password for user student.
Changing password for student
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

2. Build a **cron** job that will execute a listing of student's home directory in ten minutes from now.

```
$ crontab -e
```

Press *Insert* and add the following:

```
57 9 * * *      ls ~
```

Press *Escape* to exit the input mode, followed by *:wq* to save the changes.

3. Launch the Firefox web browser by clicking the globe icon on the menu bar at the top of the screen. Complete the web page at <http://server1/cgi-bin/roster.cgi>.
4. Open a second terminal, and SSH to localhost as "visitor" with the password of "password". Do not change this user's password. Exit the SSH session.

```
$ ssh visitor@localhost
... output truncated ...
visitor@localhost's password:
$ exit
```

5. In one of your terminal windows, use the **su** to become "root" with the password of "redhat". Use the **ifconfig** command to determine your IP address.

```
$ su -
Password:
# ifconfig
eth0  Link encap:Ethernet HWaddr 00:02:8A:AA:3B:4B
      inet addr:192.168.0.X Bcast:192.168.0.255 ...
... output truncated ...
```

6. Add two of your classmates as users on your system. Give them the password of "unpriv".

```
# useradd alice; passwd alice
... output truncated ...
# useradd bob; passwd bob
... output truncated ...
```

7. Have the two new users SSH to your station. Login to two other stations on the network.

```
$ ssh carol@stationY
... output truncated ...
carol@stationY's password:
$ exit
```

8. Use **ftp** to connect to `server1` using the user name *anonymous*. No password is needed.

```
$ ftp server1
Connected to server1.example.com
220 (vsFTPD 2.0.3)
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (server1:student): anonymous
331 Please specify the password.
Password: [Enter]
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

In the `pub` directory, download `TRANS.TBL`.

```
ftp> cd pub
250 Directory successfully changed.
ftp> get getme
local: getme remote: getme
... output truncated ...
13670 bytes received in 0.053 seconds (2.5e+02 Kbytes/s)
ftp> bye
221 Goodbye.
$ ls -l getme
-rw-rw-r-- 1 student student 13 Jan  7 23:54 TRANS.TBL
```

9. Use **vi** to add your name as the first line of this file.

```
$ vi getme
```

Insert your name to the first line of the file, and save the changes.

10. Use **SCP** to send the edited file to the systems where you have a login account.

```
$ scp ~/getme alice@stationY:~
... output truncated ...
getme                                100%   13B   13.4B/s   00:00
```

11. As root, verify that you have received a file from the users with an account on your system.

```
# head -1 /home/alice/getme
```

Alice N. Wonderland

12.

As student, use **mutt** to view the output of your cron job.

```
$ mutt
```

```
/home/student/Mail does not exist. Create it? ([yes]/no):
```

There should be an entry from Cron Daemon. Highlight the entry and press **Enter**. Once viewed, press **d** to delete, then **q** followed by **Enter** to quit.

## Unit 2

# Kernel Services

2-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



## Objectives

Upon completion of this unit, you should be able to:

- Understand kernels and modules
- Understand /proc
- Use the sysctl utility and /etc/sysctl.conf
- Discuss /dev
- Understand udev

2-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Kernel Images and Variants

- Architectures supported: x86, x86\_64, IA64/Itanium, PowerPC64, s390x.
- Three kernel versions available for x86:
  - Regular: one or more processors but 4GB of RAM or less
  - PAE: multiple processors and up to 16G of RAM
  - Xen: needed for virtualization
- Kernels always installed under `/boot/vmlinuz-*`

2-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The general idea behind providing a variety of kernel versions is that the more features are built in, the more overhead it will bring. For 32bit x86, there is a great deal of difference between the kernels.

The standard kernel is usually installed by default. It supports multiple processors, a feature known as Symmetric MultiProcessing (SMP). Memory support is limited to 4 GB physical memory though the amount of *virtual* memory could be larger through the use of swap space. The memory limit per process is 3 GB.

There is no 4G/4G ("hugemem") kernel in Red Hat Enterprise Linux 5. Former users of kernel-hugemem in Red Hat Enterprise Linux 4 are being encouraged to upgrade to the 64-bit x86-64 processor architecture when switching to Red Hat Enterprise Linux 5. With the x86-64 kernel, up to 256Gb of memory and 64 processors are supported. The per-process address space limit is also expanded from 3Gb to 512Gb.

The Xen kernel, (i.e. the kernel-xen RPM package) contains the Xen-enabled kernel for both the host and guest operating systems as well as the hypervisor. Xen is a virtual machine that can securely run multiple operating systems in their own sandboxed domains.

With kernel-xen, each domain is limited to 16GB of RAM. However, machine may have up to 64GB total. In other words, the *hardware* may have 64 GB of RAM, if you configure Dom0 to use only 16 GB of RAM and create three DomU using only 16 GB of RAM each. Such a configuration would use all 64Gb of ram on the system, and keep within supported limits for x86 kernels.

# Kernel Modules

- Modules are small kernel extensions that may be loaded and unloaded at will
- Can implement drivers, filesystems, firewall, and more
- Are located under `/lib/modules/$(uname -r)/`
- Compiled for a specific kernel version and are provided with the kernel RPM.
- Third party modules may be added

2-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A kernel module is an optional portion of kernel code that may be loaded after the kernel has been initialized. Only a few modules that are essential to all systems are included directly into the kernel. Dynamic modules that are needed at boot time are loaded by grub into the initial ram disk (initrd). Other modules may be loaded as needed later and are found in the `/lib/modules/` directory.

Modules are used for several reasons:

- Reduced memory footprint: memory is not used for drivers that are not required.
- Flexibility: modules may be added to the system after it has been installed. These modules are often called third-party drivers.
- Maximizing uptime: a module may be unloaded and reloaded as many times as wanted with no reboot.

## Kernel Module Utilities

- **lsmod** provides a list of loaded modules
- **modprobe** can load and unload modules
- **modinfo** displays information about any available module
- `/etc/modprobe.conf` used for module configuration:
  - Parameters to pass to a module whenever it is loaded
  - Aliases to represent a module name
  - Commands to execute when a module is loaded or unloaded

2-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **lsmod** utility provides a list of all loaded modules, along with the corresponding size and usage count.

```
# lsmod | grep usb_storage
usb_storage          72609  1
```

Modules may be loaded or unloaded using the **modprobe** utility. Unlike **insmod**, **modprobe** will automatically load dependencies and located the module on the filesystem. Loading a module is as easy as:

```
# modprobe usb_storage
```

To remove a currently loaded module, use **modprobe -r**:

```
# modprobe -r usb_storage
```

Note that a module may only be removed if it is not currently in use.

The **modinfo** command can be passed a module name or filename. It will display the associated information, such as the author's name, license, description, module version, dependencies, parameters, etc.

```
# modinfo usb_storage
filename:
/lib/modules/2.6.18-1.2747.el5/kernel/drivers/usb/storage/usb-storage.ko
license:      GPL
description:  USB Mass Storage driver for Linux
author:      Matthew Dharm
srcversion:   2AA118E385318B2C39E10D3
depends:      scsi_mod
vermagic:    2.6.18-1.2747.el5 SMP 686 REGPARM 4KSTACKS gcc-4.1
parm:        delay_use:seconds to delay before using a new device (uin
```

The `/etc/modprobe.conf` configuration file contains persistent settings that apply to modules commonly loaded on the system. Existing settings were added at install time, but new lines may also be added as needed:

```
alias eth0 airo
    alias snd-card-0 snd-intel8x0
    options snd-intel8x0 index=0
```

## Kernel Configuration With `/proc`

- `/proc` used to get or set kernel configuration
- Virtual filesystem: files not stored on hard disk
- Entries not persistent: modifications get reinitialized after a reboot
- Used to display process information, memory resources, hardware devices, kernel memory, etc.
- Can be used to modify network and memory subsystems or modify kernel features
- Modifications apply immediately

2-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

`/proc` is a virtual filesystem that provides detailed information about the kernel, hardware and running processes. Most files under `/proc` can be viewed with the **cat** command, though a couple use null characters to separate contents and are better viewed with the **strings** command. There isn't any real consistency to the format of data available through `/proc`. Some entries are easy to read and interpret, some are only useful to kernel developers and others are clearly not meant to be read by a human.

## /proc Examples

- Read-only files:
  - /proc/cpuinfo
  - /proc/1/\*
  - /proc/partitions
  - /proc/meminfo
- Read-Write entries under /proc/sys/:
  - /proc/sys/kernel/hostname
  - /proc/sys/net/ipv4/ip\_forward
  - /proc/sys/vm/drop\_caches
  - /proc/sys/vm/swappiness

2-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Some Interesting /proc entries:

/proc/<PID>  
/proc/cmdline  
/proc/cpuinfo  
/proc/mdstat  
/proc/meminfo  
/proc/swaps  
/proc/modules  
/proc/mounts  
/proc/net  
/proc/partitions  
/proc/version  
/proc/sys/kernel/hostname  
/proc/sys/net/ipv4/  
ip\_forward  
/proc/sys/vm/drop\_caches  
  
/proc/sys/vm/swappiness

Information on running processes (**ps**, **top**)  
Boot time options  
processor information  
software RAID information (**mdadm**)  
system memory usage (**free**, **vmstat**)  
system memory usage (**free**, **vmstat**)  
dynamically loaded modules (**lsmod**)  
mounted filesystems (**mount**)  
network activity and configuration (**ifconfig**, **netstat**)  
block devices known to the kernel  
version of the linux kernel (**uname**)  
System hostname  
IP Forwarding (on or off)

Writing a 1 forces the kernel to free up some memory from caches.

Indicates how aggressively memory will be swapped out to the swap device (number between 0 and 100).

# sysctl : Persistent Kernel Configuration

- **sysctl** adds persistence to `/proc/sys` settings
- Statements added to `/etc/sysctl.conf` automatically reflected under `/proc` after a reboot.
- Configuration maintained or monitored using the **sysctl** command:
  - List all current settings: **sysctl -a**
  - Reload settings from `sysctl.conf`: **sysctl -p**
  - Set a `/proc` value dynamically: **sysctl -w net.ipv4.ip\_forward=1**

2-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

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. Because of this, a great many `sysctls` exist, many of which are not documented in a way useful to an administrator. This is generally okay, because there's a good chance that the parameter would not be useful anyway. Commonly useful parameters will be documented online, in `kernel-doc` or in this and other Red Hat courses.

The `sysctl` command can be used to view and set kernel parameters. To get a list of all parameters and their values:

```
# sysctl -a
```

To set a parameter, `net.ipv4.tcp_syncookies` in this case, we could use:

```
# sysctl -w net.ipv4.tcp_syncookies=1
```

To make this setting permanent, we would want to add the parameter to `/etc/sysctl.conf`.

Once this is done, the new configuration file could be synchronized with the kernel by using:

```
# sysctl -p
```



# Accessing Drivers Through /dev

- Files under /dev are used to access drivers
- Reading from and writing to those files are valid operations:
  - Read from serial port: `cat /dev/ttyS0`
  - Write to serial port: `echo "Message" > /dev/ttyS0`
- Three file attributes determine which driver to access:
  - Device type (character or block)
  - Major number
  - Minor number

2-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Device nodes provide access points for device drivers and map standard file permissions onto that access. The two types of device nodes are block and character. A quick description of these would be that block devices handle data storage while character devices are more suitable for data streams. A more precise description would be that block devices use kernel buffered IO and character devices do not use buffers.

Reading from a serial port, for instance, can be easily accomplished using the following command:

```
# cat /dev/ttyS0
```

Writing a word to the same serial port, on the other hand, would require:

```
echo "Message" > /dev/ttyS0
```

Device special files located under /dev are all tagged with three attributes: the device type (character or block), the Major number and the Minor number. The `ls -l` command can be used to display those attributes:

```
# ls -l /dev/ttyS0
crw-rw-r-- 1 root uucp 4, 64 Feb  4 16:24 /dev/ttyS0
```

In the previous example, the 'c' at the beginning of the line indicates a character device. Notice that there is no file size: this field was replaced with 4, 64, which indicates a major number equal to 4 and a minor number equal to 64. The major number determines which driver to access, whereas the minor number allows the driver to differentiate between similar physical devices.

Similarly, a block device such as a USB disk would look as follows:

```
# ls -l /dev/sda1
brw-r----- 1 root disk 8, 1 Feb  4 16:12 /dev/sda1
```

# Device Node Examples

- **Block Devices**
  - `/dev/hda`, `/dev/hdc` - IDE hard disk, CDROM
  - `/dev/sda`, `/dev/sdb` - SCSI, SATA, or USB Storage
  - `/dev/md0`, `/dev/md1` - Software RAID
- **Character Devices**
  - `/dev/tty[0-6]` - virtual consoles
  - `/dev/null`, `/dev/zero` - software Devices
  - `/dev/random`, `/dev/urandom` - random Numbers

2-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

IDE drives are labeled `/dev/hd*`. `/dev/hda` is the master drive on the first IDE controller. `/dev/hdb` is the slave on the first controller. `/dev/hdc` is the master on the second controller and `/dev/hdd` is the slave on the second controller. Most IDE based systems use `/dev/hda` as the primary hard disk with the CD/DVD drive connected as `/dev/hdc`. If an additional drive is added, it will normally be `/dev/hdd` to avoid overloading the first IDE controller.

SCSI storage devices can be a number of different things: plain SCSI hard disks, Serial ATA, SAN storage devices, USB devices. The key to understanding this is that SCSI is a protocol, not a type of device. So even with something like a USB storage device, developers chose to stick with an established standard for the interface. One particular challenge with SCSI devices is how they are named: which drive will be `/dev/sda`, which is `/dev/sdb` and such. The naming of devices occurs in the order the devices are discovered. There is no guarantee that the drive that is `/dev/sdb` now won't become `/dev/sdc` after a reboot. To get around this issue, various labeling schemes are used for filesystems and swap space to avoid addressing the device directly through the device node.

One common class of character device is the terminal. Basically, whenever you're working at a shell, you're using a terminal device. The command **who am i** can be used to check which device you're using. Using just **who** will show what terminals are in use.

The virtual consoles provide text logins through the kernel's VGA driver and are identified as `/dev/tty` followed by a number. There are normally six virtual consoles that provide logins.

Pseudo terminals are used by programs like **gnome-terminal** and the ssh server. These terminals are dynamically created by a special virtual filesystem, `devpts`, under the `/dev/pts/` directory.

## Managing /dev With udev

- udev manages files stored under /dev/
- Files are only created if corresponding device is plugged in
- Files are automatically removed when device is disconnected
- udev statements under /etc/udev/rules.d/ determine:
  - Filenames
  - Permissions
  - Owners and groups
  - Commands to execute when a new device shows up

2-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (918) 754 3700.

Linux distributions used to ship with thousands of files under the /dev directory. Although this worked, it certainly was not elegant to provide every single device special file known to mankind even if the corresponding device would never be added to your system.

A newer, better, and more efficient system was then added to replace static files under /dev. This system is called *udev*. It consists in a series of utilities and configuration files which provide rules that apply whenever a device is connected to the system and detected by the kernel.

udev makes it possible to create or remove files on the fly, when the corresponding device is plugged or disconnected. It also lets system administrator add rules in order to modify default names and permissions used under /dev.

Rules are located under /etc/udev/rules.d/. Red Hat Enterprise Linux ships with a default set of rules that should work in most cases. When adding your own rules, we suggest adding a new file with your rules in it, rather than editing the existing files. udev.

## Adding Files Under /dev

- The right way to add a /dev entry involves udev:

- Create a new file under /etc/udev/rules.d/
- Insert a statement such as:

```
KERNEL=="sda", NAME="usbkey" , SYMLINK="usbstorage"
```

- This creates a device file named usbkey and a symlink named usbstorage next time /dev/sda gets plugged.
- Files can be added manually with **mknod**:

```
mknod /dev/usbdevice b 8 0
```

- **mknod** is not persistent!

2-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Adding a device special file under /dev can be done in two ways: using **mknod** or by adding a new rule to udev.

The udev method involves adding a file under /etc/udev/rules.d/. This file should contain one or several statements that comply with the udev syntax. A statement such as the following would replace the default /dev/sda file with /dev/usbkey. A symbolic link called /dev/usbstorage would also be created. All these files would get automatically created when the device gets plugged in, and removed when it is disconnected.

Alternatively, a file can be quickly created under /dev using the **mknod** command. This would require knowledge about the device type, major and minor numbers and work as follows:

```
# mknod /dev/usbdevice b 8 0
```

The MAKEDEV utility can also be used as an easier replacement to **mknod**. In both cases, the new file would not persist after a reboot.

## Exploring Hardware Devices

- A snapshot of all connected devices is maintained by HAL: Hardware Abstraction Layer
- **hal-device** lists all devices in text mode.
- **hal-device-manager** displays all devices on a graphical window.
- **lspci** and **lsusb** list devices connected to the PCI and USB buses, respectively.
- The `/proc` and `/sys` filesystems also contain bus and device specific information.

2-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Hardware devices can be monitored in a number of different ways. The `/proc` filesystem has historically been the main method, with files such as `/proc/devices`, `/proc/interrupts`, `/proc/iomem` and `/proc/ioports`. Buses such as the PCI and USB bus are also exposed through the `/proc/bus/` directory.

To make the `/proc` entries more readable, utilities such as **lspci** and **lsusb** are also provided.

More recently, however, a new layer has been provided to expose hardware information: HAL (Hardware Abstraction Layer). HAL continuously maintains a snapshot of all hardware devices currently connected to the system. This snapshot may be monitored in text mode using the **hal-device** command, or in graphical mode with the **hal-device-manager** application.

Much of the information provided through HAL can also be accessed from the `/sys` filesystem.

## End of Unit 2

- Questions and Answers
- Summary
  - lsmod, modprobe
  - sysctl, /proc
  - udev, /dev

2-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Lab 2

# Configuring the kernel

---

Goal:

Develop skills tuning the `/proc` filesystem.

Gain some experience working with device special files and modules.

Use the tools available to explore hardware resources.

## Sequence 1: Turning off ping responses

**Scenario:** You want to reduce the exposure of a critical system. One of your strategies is to “hide” it from easy discovery by ICMP ECHO requests.

**Deliverable:** A system that does not respond to **ping**.

**Instructions:**

1. Configure your system, so that it does not respond to any **ping** request. This configuration should survive a reboot.

Hint: Install the `kernel-doc` package and check the kernel documentation on `/usr/share/doc/kernel/Documentation/networking/ip-sysctl.txt`.

### 2. MANDATORY CLEANUP

- a. Comment out or remove `net.ipv4.icmp_echo_ignore_all=1` from `/etc/sysctl.conf`
- b. Remember that changing this file does not affect the system's current configuration, so you will want to undo your change directly as well:

```
# echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

This is to prevent other things from breaking during the week and help preserve your and your instructor's sanity.



## Optional Sequence 2: Creating a file persistently under /dev/

**Scenario:** You want to make sure the /dev/myusbdisk filename is available after a reboot and can be used to mount a USB device.

**Deliverable:** A system that provides /dev/myusbdisk automatically after a reboot.

**System Setup:** System running in runlevel 5.

### Instructions:

1. Modify the udev subsystem in such a way that /dev/myusbdisk gets automatically created at boot time.
2. Reboot the system.  
  
Plug a USB key to your system and verify that you now have a file named /dev/myusbdisk.
3. **MANDATORY CLEANUP:** Remove the file you have created under /etc/udev/rules.d/ and unplug the USB device.

## Sequence 1 Solutions

1. Configure your system, so that it does not respond to any **ping** request.
  - a. Check the present value of `/proc/sys/net/ipv4/icmp_echo_ignore_all`  

```
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```

It should be currently set to zero which means your system will respond normally to pings.
  - b. Change the value of `/proc/sys/net/ipv4/icmp_echo_ignore_all` to a 1 which will prevent other hosts from successfully pinging your host while not affecting your ability to ping them. Verify your work.  

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all  
# cat /proc/sys/net/ipv4/icmp_echo_ignore_all
```
  - c. Now test pinging `server1.example.com`. Pressing **Ctrl-C** will stop the **ping** command and display some statistics for you. You should have been able to ping `server1`.
  - d. Next have someone else try pinging your station. They should not receive any responses back from your system. Alternatively, try to ping your own network address. This should not work either.
  - e. Now reboot your system and try to ping your station again. What happened? Why?
  - f. Remember that changes to the `/proc` filesystem are temporary and if you want them to persist across reboots you need to put an entry in `/etc/sysctl.conf`. Edit `/etc/sysctl.conf` and put the following line at the bottom:  

```
net.ipv4.icmp_echo_ignore_all=1
```
  - g. To activate this change run:  

```
# sysctl -p
```
  - h. Check the value in `/proc`. If it is not set to a 1 then recheck the previous two steps. Next reboot your system and check the value in `/proc` again.
2. MANDATORY CLEANUP
  - a. Comment out or remove `net.ipv4.icmp_echo_ignore_all=1` from `/etc/sysctl.conf`
  - b. Reboot the system.

This is to prevent other things from breaking during the week and help preserve your and your instructor's sanity.

## Optional Sequence 2 Solutions

1. Modify the udev subsystem in such a way that `/dev/myusbdisk` gets automatically created at boot time.

Create a file named `/etc/udev/rules.d/99-usb.rules` and insert the following statement in it:

```
KERNEL=="sdb1", NAME="myusbdisk"
```

*Note:* Systems with IDE harddrives use `sda` for the first USB disk.

2. Reboot the system:

```
init 6
```

Plug a USB key to your system and verify that you now have a file named `/dev/myusbdisk`.

```
ls -l /dev/myusbdisk
```

3. **MANDATORY CLEANUP:** Remove the file you have created under `/etc/udev/rules.d/` and unplug the USB device:

```
rm /etc/udev/rules.d/99-usb.rules
```

## Unit 3

# Filesystem Management

3-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Create partitions
- Create and mount a filesystem
- Unmount a filesystem
- Increase virtual memory

3-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## fdisk

- View and manage partition tables
- List partition table from command line
  - **fdisk -l**
- Manage partition table from interactive interface
  - **fdisk /dev/sda**
- **partprobe**
- **cat /proc/partitions**

3-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

### *View partition table*

There are two ways to view the partition table:

```
[root@stationX ~]# cat /proc/partitions
```

will allow us to see the partition tables as viewed by the kernel. To view the partition table physically stored on a particular disk:

```
[root@stationX ~]# fdisk -l /dev/hda
```

### *Manage partition table*

To manage the partition table utilizing **fdisk**, we must enter in to the interactive user interface of **fdisk**. This can be done by entering:

```
[root@stationX ~]# fdisk /dev/hda
```

From the interactive interface, there are a number of single character commands that can be entered:

```
m - display the menu/help
p - print the working table
n - create a new partition
t - change a partition id or type value
d - delete a current partition
w - save changes to disk and exit
q - quit without saving
```

It is important to remember that **fdisk** can not modify existing partitions. It can delete an existing partition and add a new partition. This means that to change a partition, it must be deleted, then added back to the table. As long as the starting point of the partition does not change, only the end point, it is possible that the data will remain intact. This is called *truncating* or *extending* a partition and should never be attempted without first backing up the data.

Changes made in **fdisk** do not take effect until they are committed with a **w** command. This means it is easy to recover from a mistake by pressing **q** and starting over.

When creating a new partition, keep in mind the way drives are organized. On an i386, x86\_64, or ia64 system, drive are limited to four *primary partitions*. The last of these primary partitions, number 4, could be set to be an *extended partition*. Within the extended partition additional partitions, called *logical partitions* could be created.

The **n** command will prompt for two entries, starting cylinder and ending cylinder. To simplify data entry, we can enter “+sizeM” as the ending cylinder (M stands for megabyte, alternatively use G for gigabyte) and **fdisk** will calculate the ending cylinder for us.

The **t** command will prompt for two entries, partition number and partition id. Be careful, the first is asking us which partition we want to modify (1-15), while the second one is asking us the two-digit hexadecimal code that represents the “kind” of data being stored on that partition. We should know the following id or type values:

- 5 - Extended
- 7 - HPFS/NTFS
- c - W95 FAT32 (LBA)
- 83 - Linux filesystem (ext2, ext3, or others)
- 82 - Linux swap
- fd - Linux RAID (used for auto detection)
- 8e - Linux LVM

It is also useful to remember that **fdisk** will present a full list of known partition types if you enter **L** when it prompts for the value.

# Making Filesystems

- **mkfs**
- **mkfs.ext2, mkfs.ext3, mkfs.msdos**
- Specific filesystem utilities can be called directly
  - **mke2fs [options] device**

3-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command for creating a filesystem is **mkfs**. This command is a front end or wrapper for various filesystem creation programs. **mkfs -t *fstype*** looks for programs that follow the naming convention **mkfs.*fstype***, and then runs the program to create the desired filesystem. If you run **mkfs** without the **-t** option, it assumes the Linux default ext2 (Second Extended) filesystem. The **mkfs.*fstype*** form of the program can also be called directly. The ext2 filesystem has an additional interface called by:

## **mke2fs [options] device**

Some useful options include:

**-b** Specify the size of data blocks in bytes. Without this option, the data block size is determined by the size of the partition. Filesystems that will contain many small files, for example, should use smaller block sizes because each file uses an entire block, even if its data is less than the size of one block. One file occupies, at least, one data block. Compare the variable size of a data block with the invariable size of the physical hard drive sector (usually 512 bytes).

**-c** Check the device for bad blocks (sectors) before creating the filesystem. This can take several hours if the partition is very large.

**-i** Specify the bytes/inode ratio. **mke2fs** creates an inode table based on the total size in bytes of a potential filesystem. Large datasets managed by a filesystem benefit from a higher ratio as there are fewer inodes, freeing more data space. Only one inode is allocated per file, while one inode may reference one or more data blocks. Inodes are 128 bytes in size, and unused inodes can occupy a lot of available space.

**-N** Override the default calculation of how many inodes are reserved for the filesystem. By default, this is based on the number of blocks and the bytes/inode ratio. The **-N** option allows you to specify the number of inodes directly. *Note:* This may be a more effective method of utilizing the partition or data space bytes/inode ratio for extremely large datasets, as the largest value passed to the **-i** option above is 8192.

**-m** Specify the percentage of reserved blocks for the superuser. This value defaults to 5%. If the filesystem being created is for a specific application, changing this percentage to zero allows the application full use of the filesystem. However, the filesystem also uses the reserved blocks to avoid *fragmentation*. Fragmentation happens when there are not enough blocks available to save a file completely within one *block group*. A highly fragmented filesystem performs poorly. The only way to remove fragmentation is to recreate the filesystem and restore the data from backup.

**-L** Set the volume label for the filesystem. This option's usefulness becomes apparent later when we discuss how filesystems are connected together on a Red Hat Enterprise Linux system.



**-j** Create an ext3 journal inode and filesystem. This is default for ext3.

# Filesystem Labels

- Alternate way to refer to devices
- Device independent
  - **e2label** *special\_dev\_file* [*fslabel*]
  - **mount** [*options*] **LABEL=fslabel** *mount\_point*
- **blkid** can be used to see labels and filesystem type of all devices.

3-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A potential problem exists when using special device files to point to file systems in that if the device is somehow relocated on the system, then the special device file that points to it will change. This most commonly happens with SCSI devices. Filesystem labels provide an alternate way to reference file systems for mounting that is not dependent on the special device file.

Two mechanisms exist to support filesystem labels. A filesystem label can be written into the superblock of ext2/ext3 filesystems using the **e2label** command:

**e2label /dev/hda7 dbdisk1**

Would create a label of **dbdisk1** on the filesystem on partition **/dev/hda7**. The command:

**e2label /dev/hda7**

will display the current filesystem label for that device. The filesystem can be mounted using the command:

**mount LABEL=dbdisk1 /mnt/data**

# Mount Points and /etc/fstab

- Configuration of the filesystem hierarchy
- Used by **mount**, **fsck**, and other programs
- Maintains the hierarchy between system reboots
- May use filesystem volume labels in the device field
- The **mount -a** command can be used to mount all filesystems listed in /etc/fstab

3-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

/etc/fstab is referenced each time the system boots to create the desired filesystem hierarchy. It consists of six fields per line for each filesystem to be connected to the tree as follows:

device	mount_point	FS_type	options	dump_freq	fsck_order
LABEL=/mnt/data	/mnt/data	ext3	defaults	0	0

**device** -- The special device file name, or filesystem label of the device to mount

**mount\_point** -- the path used to access the filesystem

**FS\_type** -- The filesystem type

**options** -- A comma-separated list of options, see **man mount**

**dump\_freq** -- Dump frequency: 1=daily, 2=every other day, etc.; 0=never dump

**fsck\_order** -- Priority: 0 = ignore, 1 = first (the root filesystem should have this value), 2-9 = second, third, etc.: filesystems that have the same number greater than 1 are checked in parallel. Network filesystems and CD-ROMs should be ignored.

Using /etc/fstab when creating new mount points: After the filesystem has been created on a device and the filesystem has been labeled, create an entry in the fstab that shows where it should be mounted. Important: After the entry in /etc/fstab has been created use **mount -a** to mount the filesystem just created and check for errors. It's better to discover an error when the system is online then to discover it later. In many cases errors in /etc/fstab will cause **rc.sysinit** to fail and boot to **sulogin** (also known as emergency mode).

During system initialization, /etc/fstab is used to create the filesystem hierarchy. Entries are parsed and used as arguments **mount**, unless the option **noauto** is present. Floppy and CD-ROM entries typically have **noauto** as an option.

/etc/fstab is also used after system initialization, when partitions are mounted manually. If the partition /dev/hda5 contains a filesystem labeled /mnt/data, the directory /mnt/data exists, and is already available, then the following mount commands would connect this filesystem to the filesystem tree:

```
[root@stationX ~]# mount /dev/hda5  
[root@stationX ~]# mount -L /mnt/data  
[root@stationX ~]# mount LABEL=/mnt/data  
[root@stationX ~]# mount /mnt/data
```

# Unmounting Filesystems

- **umount** [*options*] *device* | *mount\_point*
- You cannot unmount a filesystem that is in use
  - Use **fuser** to check and/or kill processes
- Use the **remount** option to change a mounted filesystem's options atomically
  - **mount -o remount,ro /data**

3-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

There are a few reasons for unmounting or disconnecting a filesystem from the root hierarchy. When you bring the system off line, reboot it, or perform filesystem maintenance, or when you use removable media, disconnect filesystems with **umount**. At shutdown or reboot time, the init scripts should do this for you automatically, but removable media and maintenance mode both frequently require manual unmounting. **umount** references `/etc/mtab`. You can run **umount -a** (only as superuser) to disconnect all filesystems, or run:

```
umount [options] device | mount_point
```

For reasons of operating system stability and protection, you cannot unmount a filesystem that has open files, file handles, or a process's CWD, or is otherwise in use. On a removable media device, the device driver attempts to lock the device. If successful, the lock is removed when the device is quiescent, when unused. This lock, or an ignored **umount** command, can be frustrating. Should you experience this, **fuser** is helpful.

**fuser** displays information about the processes using a filesystem. After determining what is acting on the filesystem, **fuser** also provides a convenient way to send signals to those processes. It can prompt you interactively for each process before sending a signal, or kill all processes acting on the filesystem, including a user's CWD entry. To display what (or who) is acting on filesystem, run:

```
fuser -v mount_point
```

To kill all actions on a filesystem, run:

```
fuser -km mount_point
```

Sometimes you need to change the options of a mounted filesystem atomically (that is, without other operations occurring during the change). For example, if the root filesystem is mounted read-only (as is common during recovery operations), and you want to edit a configuration file on the root filesystem, you must mount the filesystem read-write before you can edit the file. If the root filesystem is on `/dev/hda5`, the following **mount** command reinitializes the root filesystem's mount in a single operation (atomically), using the new options:

```
mount -o remount,rw /dev/hda5 /
```

# Handling Swap Files and Partitions

- Swap space is a supplement to system RAM
- Basic setup involves:
  - Creating a swap partition or file
  - Writing special signature using **mkswap**
  - Adding appropriate entries to `/etc/fstab`
  - Activating swap space with **swapon -a**

3-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Setting up a swap partition

Use a partitioning program to add a partition. Set the partition id type to 0x82. Create the signature needed on the partition using **mkswap**:

**mkswap /dev/hda6**

Add an entry for the swap to `/etc/fstab`. It will look similar to the following:

```
/dev/hda6 swap swap defaults 0 0
```

Activate the swap partition using **swapon -a** (which reads `/etc/fstab` and turns on all swap entries it lists).

Check the swap partition's status using **swapon -s**.

## Setting up a swap file

Use the following to create a file, where count X defines the file size in kilobyte blocks:

**dd if=/dev/zero of=/swapfile bs=1024 count=X**

Run **mkswap** to create the signature. The swap file can also be activated manually with **swapon**, or added entry to `/etc/fstab`:

```
/swapfile swap swap defaults 0 0
```

## End of Unit 3

- Questions and Answers
- Summary
  - Creating partitions
  - Creating and mounting a filesystem
  - Unmounting filesystems
  - mkswap, swapon

3-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[ctraining@redhat.com](mailto:ctraining@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 3

## Creating Filesystems

---

Goal: Understand partitioning and the creation of filesystems.

System Setup: Workstation installation with free space on the hard drive.



## Sequence 1: Create a new filesystem

**Scenario:** A new application needs to be installing in the /opt directory. For recovery reasons, /opt needs to be a dedicated partition. The new /opt partition needs to be available after a reboot.

**Deliverable:** A system with a separate partition mounted to /opt.

### Instructions:

1. Use **fdisk -l** to locate information about your hard disk. You will see either a /dev/sda or /dev/hda disk. Partition sizes are given in cylinders, which vary in size depending on the drive's construction. Notice the cylinder count in the heading and the ending cylinder of the list.
2. Use **fdisk** to add a new *logical* partition that is 256M in size. Execute **fdisk -l** to verify that the partition was created.
3. While writing the partition table, the kernel encountered an error, and requested a reboot. This is normal, but unnecessary. View the contents on /proc/partitions. It should only reflect the original partitions. Execute **partprobe** and review /proc/partitions.
4. Create an ext3 filesystem on the new partition and assign the label opt. Modify /etc/fstab such that the partition will mount to /opt at boot time.
5. Test your changes with **mount -a**.

## Sequence 2: Creating a new swap partition

**Scenario:** The system's on-disk swap space need to be increased after an memory upgrade. This a new partition needs to be created, without effect other existing partitions or mount points. All swap partitions should become active after each reboot.

**Deliverable:** A system with two swap partitions.

### Instructions:

1. Use **fdisk** to create an additional swap partition of 2G. Ensure the partition is tagged as a type 82 Linux swap.
2. Create the swap signature. Remember that you can not label a swap partition with **e2label**. Refer to **man mkswap** to determine how to label the partition.
3. Configure `/etc/fstab` to activate the swap partition at boot time.
4. Verify swap's status with **swapon -s**. Activate the new partition with **swapon -a**. If the command completed without error, verify the swap status again.

## Sequence 1 Solutions

1. Use **fdisk -l** to locate information about your hard disk. You will see either a `/dev/sda` or `/dev/hda` disk. Partition sizes are given in cylinders, which vary in size depending on the drive's construction. Notice the cylinder count in the heading and the ending cylinder of the list.

```
# fdisk -l
```

```
Disk /dev/sda: 3221 MB, 3221225472 bytes
255 heads, 63 sectors/track, 391 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	268	2048287+	83	Linux
/dev/sda3		269	333	522112+	82	Linux swap / Solaris

In this example, 391 is the total cylinder count for the device, and 333 is the last allocated cylinder.

2. Use **fdisk** to add a new *logical* partition that is 256M in size. Ensure you are editing the disk's partition table, not the first partition.

```
# fdisk /dev/sda
```

Most PC based systems require the fourth partition to be an *extended* partition that spans the total free space of the drive. Other *logical* partitions are created within the extended partition.

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

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)  e
```

```
Selected partition 4
```

```
First cylinder (334-391, default 334): [Enter]
```

```
Using default value 334
```

```
Last cylinder or +size or +sizeM or +sizeK (334-391, default 391): [Enter]
```

```
Using default value 391
```

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

```
Disk /dev/sda: 3221 MB, 3221225472 bytes
255 heads, 63 sectors/track, 391 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	268	2048287+	83	Linux
/dev/sda3		269	333	522112+	82	Linux swap / Solaris

```
/dev/sda4      334   391   465885    5   Extended
```

In this example, the fifth partition will use the same starting cylinder as the extended partition, but will use only a portion of the extended partition's space.

Command (m for help): **n**

First cylinder (334-391, default 334): **[Enter]**

Using default value 334

Last cylinder or +size or +sizeM or +sizeK (334-391, default 391): **+256M**

Command (m for help): **p**

... output truncated ...

```
/dev/sda4      334   391   465885    5   Extended
```

```
/dev/sda5      334   365   257008+   83   Linux
```

Commit the changes and verify that they have been saved.

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.

Syncing disks.

```
# fdisk -l
```

... output truncated ...

```
/dev/sda4      334   391   465885    5   Extended
```

```
/dev/sda5      334   365   257008+   83   Linux
```

3. While writing the partition table, the kernel encountered an error, and requested a reboot. This is normal, but unnecessary. View the contents on `/proc/partitions`.

```
# cat /proc/partitions
```

```
major minor #blocks name
```

... output truncated ...

```
202      3      522112 sda3
```

The table will only reflect the original partitions. Execute **partprobe** and review `/proc/partitions`.

```
# partprobe
```

```
# cat /proc/partitions
```

```
major minor #blocks name
```

... output truncated ...

```
202      3      522112 sda3
```

```
202      4      1 sda4
202      5      257008 sda5
```

4. Create an ext3 filesystem on the new partition and assign the label opt.

```
# mke2fs -j /dev/sda5
... output truncated ...
# e2label /dev/sda5 opt
# blkid | grep /dev/sda5
/dev/sda5: LABEL="opt" UUID="..." SEC_TYPE="ext2" TYPE="ext3"
```

Alternately, a quick check of **man mke2fs** would indicate that labeling can be done during filesystem creation:

```
# mke2fs -j /dev/sda5 -L opt
mke2fs 1.39 (29-May-2006)
Filesystem label=opt
... output truncated ...
# blkid | grep /dev/sda5
/dev/sda5: LABEL="opt" UUID="..." SEC_TYPE="ext2" TYPE="ext3"
```

Modify `/etc/fstab` such that the partition will mount to `/home` at boot time.

```
# vi /etc/fstab; grep "opt" /etc/fstab

LABEL=opt      /opt      ext3      defaults    0 0
```

5. Test your changes with **mount -a**.

```
# mount | grep "opt"
# mount -a
# mount | grep "opt"
/dev/sda5 on /opt type ext3 (rw)
```

Troubleshoot any errors to ensure the partition mount properly.

## Sequence 2 Solutions

1. Use **fdisk** to create an additional swap partition of 2G. Ensure the partition is tagged as a type 82 Linux swap.

```
# fdisk /dev/sda
```

```
Command (m for help): p
... output truncated ...
```

/dev/sda3	269	333	522112+	82	Linux swap / Solaris
/dev/sda4	334	1151	465885	5	Extended
/dev/sda5	334	365	257008+	83	Linux

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

```
First cylinder (366-391, default 366): [Enter]
```

```
Using default value 366
```

```
Last cylinder or +size or +sizeM or +sizeK (366-391, default 391): +2G
```

```
Command (m for help): p
... output truncated ...
```

/dev/sda6	366	588	2048287	83	Linux
-----------	-----	-----	---------	----	-------

```
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): p
... output truncated ...
```

/dev/sda6	366	588	2048287	82	Linux swap / Solaris
-----------	-----	-----	---------	----	----------------------

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

Commit the changes with the **w**. You will see a message indicating that you must reboot. Use **partprobe** instead.

```
# cat /proc/partitions
... output truncated ...
202      5      257008 sda5
```

```
# partprobe
```

```
# cat /proc/partitions
... output truncated ...
202      5      257008 sda5
202      6      2048287 sda6
```

2. Create the swap signature. Remember that you can not label a swap partition with **e2label**. Refer to **man mkswap** to determine how to label the partition.

```
# mkswap /dev/sda6 -L newswap
```

```
Setting up swapspace version 1, size = 139792 kB
LABEL=newswap, no uuid
# blkid
/dev/sda3: TYPE="swap" LABEL="SWAP-sda3"
... output truncated ...
/dev/sda6: TYPE="swap" LABEL="newswap"
```

3. Configure `/etc/fstab` to activate the swap partition at boot time. Add a line similar to the following:

```
LABEL=newswap      swap      swap      defaults    0 0
```

4. Verify swap's status with **swapon -s**. Activate the new partition with **swapon -a**. If the command completed without error, verify the swap status again.

```
# swapon -s
Filename      Type      Size      Used      Priority
/dev/sda3     partition 522104    56        -1
# swapon -a
# swapon -s
Filename      Type      Size      Used      Priority
/dev/sda3     partition 522104    56        -1
/dev/sda6     partition 2048287   0         -2
```

# Unit 4

## User Administration

4-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2984 or +1 (919) 754 3700.



## Objectives

Upon completion of this unit, you should be able to:

- Understand user administration
- Use special permissions
- Set up file system quotas

4-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Modifying User Accounts

- Most user settings are in `/etc/passwd`:
  - username, UID, GID (user private group), comment, home directory, login shell
- Other files:
  - `/etc/group`, `/etc/shadow`
- To change user information you can:
  - Edit the files by hand
  - Use **usermod** `[options] username`
  - Use **system-config-users**

4-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

In addition to hand-editing `/etc/passwd`, you may use **usermod** to change account information. The following options are available:

<b>-c comment</b>	Change the comment field. This is often the users full name.
<b>-d home dir</b>	Change the home directory.
<b>-e expire date</b>	Set date on which the account will expire and be disabled.
<b>-g group</b>	Change the initial login group
<b>-G group, [...]</b>	A comma separated list of supplementary groups for the user.
<b>-l login name</b>	Change the login name.
<b>-s shell</b>	Change the login shell.
<b>-u uid</b>	Change the login ID.
<b>-p password</b>	Change the string in the password field
<b>-L</b>	Lock the password. This renders the account unusable.
<b>-U</b>	Unlock the password.

Unfortunately, **usermod** assumes that a user would never be in more than a single supplemental group at a time. Thus, when using the **-G** option, you must list all the user's groups-- even the ones that aren't changing! By adding the **-a** option before the **-G**, the user will be added to the specified groups, without removing them from existing groups.

```
[root@stationX ~]# grep student /etc/group
mail:x:12:student
student:x:500:
[root@stationX ~]# usermod -G news student
[root@stationX ~]# grep student /etc/group
news:x:13:news,student
student:x:500:
[root@stationX ~]# usermod -a -G mail student
[root@stationX ~]# grep student /etc/group
mail:x:12:student
```

news:x:13:news,student  
student:x:500:

# Group Administration

- Entries added to `/etc/group` and `/etc/gshadow`
  - **groupadd**
  - **groupmod**
  - **groupdel**

4-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

New groups may be created by hand-editing the file `/etc/group` or by using **groupadd**. The basic syntax for **groupadd** is very simple:

```
[root@stationX] # groupadd groupname
```

**groupdel** is used in a similar fashion to remove groups:

```
[root@stationX] # groupdel groupname
```

**groupmod** can be used, among other things, to change the name of a group:

```
[root@stationX] # groupmod -n newname oldname
```

For example, if several users are members of the employee group and a root user issues the following command, the group will be changed to staff and all the same members will remain:

```
[root@stationX] # groupmod -n staff employee
```

The syntax of `/etc/group` is as follows, one group per line:

```
gurus:x:501:joshua,dax,bryan,chris,heather,jon
```

The first field is the name of the group. The second field is the group password, or an "x" when using shadow passwords. The third field is the unique group ID. The fourth field is a comma-separated list of group members.

In order to avoid using a GID within the range typically assigned to users and their private groups, use the **-r** option:

```
[root@stationX] # groupadd -r groupname
```

This starts the group IDs at 101 and will increase them up to GID 499.

**gpaswd** can be used to define group members in `/etc/group`, group administrators, and to create or change group passwords in `/etc/gshadow`, if desired.

# Password Aging Policies

- By default, passwords do not expire
- Forcing passwords to expire is part of a strong security policy
- Modify default expiration settings in `/etc/login.defs`
- To modify password aging for existing users, use the **chage** command
  - **chage** [*options*] *username*

4-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

By default passwords do not expire. This means that it is possible for a user to have the same password indefinitely. This situation is not very secure, because if a password has leaked, or been compromised, it will remain so forever. This can be adjusted in the `/etc/login.defs` file.

The **chage** command is used to set up password aging. You may set the maximum amount of time that a password is considered valid before the system will force the user to change his password. The security policy of an organization will generally define the amount of time between password changes. It is also important to set the minimum amount of time that a password must be used before it can be changed. This prevents users from changing their password when required to by the system, and then changing it right back to the old value.

```
[root@stationX]# chage [options] username
```

Common options used with the **chage** command:

<b>-m</b>	minimum days between password changes
<b>-M</b>	maximum days between password changes
<b>-I</b>	number of days inactive since password expired before locking account
<b>-E date</b>	expire the account on this date (YYYY-MM-DD format)
<b>-W</b>	number of days before a required change to start warnings
<b>-l</b>	list the settings

**chage** *username* without options will let you change the settings interactively.

# Deleting Accounts

- To remove a user either:
  - Manually remove the user from `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow`, `/var/spool/mail`
  - Use **userdel** *username*
- Neither option removes data files owned by the user
- To delete the home directory, use **userdel -r**

4-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **userdel** deletes a user account from the system. It may be prudent to lock the user's account first with **usermod -L**, and to delay deletion of the user's account until you are sure that none of the files in the user's home directory are still needed.

When deleting an account with **userdel**, you should consider using the **-r** option, which recursively deletes the user's home directory. Deleting accounts without deleting the associated home directories may cause issues with file ownership when future users are added to the system.

To ensure there are no orphaned files on the system, it is best to use **find** to search the filesystem.

```
[root@stationX ~]# find / -uid 500
find: /proc/1415/task/1415/fd/4: No such file or directory
find: /proc/1415/fd/4: No such file or directory
/data/toyota.txt
/data/citron.txt
/tmp/kia.txt
/tmp/ford.txt
```

## SGID Directories

- Used to create a collaborative directory
- Normally, files created in a directory belong to the user's default group
- When a file is created in a directory with the SGID bit set, it belongs to the same group as the directory

4-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When a file is created in a directory, it belongs to the primary group of the user that created the file.

```
[root@stationX]# mkdir /shared
[root@stationX]# ls -ld /shared
drwxr-xr-x 2 root root 4096 Feb 27 11:28 /shared
[root@stationX]# touch /shared/first
[root@stationX]# ls -l /shared
-rw-r--r-- 1 root root 0 Feb 27 11:33 first
```

However, if the *setgid* (SGID or Set Group ID) bit is set for the directory, new files that are created in the directory have their group ownership set to the same group ownership as the directory. This provides a mechanism to allow one level of access to users, who are members of the same group that owns the directory while allowing a different level of access to non-member users of files in the directory.

Recall that permissions can be set with **chmod**:

```
[root@stationX]# chmod g+s directory
```

This sets the SGID bit without affecting current permissions. Alternately:

```
[root@stationX]# chmod 2770 directory
```

The above syntax sets the SGID bit and gives read, write, and execute permissions to the owner of the directory and members of the group whose ownership is on that directory.

```
[root@stationX]# chgrp mail /shared
[root@stationX]# ls -ld /shared
drwxr-xr-x 2 root mail 4096 Feb 27 11:28 /shared
[root@stationX]# chmod g+s /shared
[root@stationX]# ls -ld /shared
drwxr-sr-x 2 root mail 4096 Feb 27 11:28 /shared
```

In this example, all future files will be created as members of the mail group.

```
[root@stationX]# touch /shared/second
[root@stationX]# ls -l /shared
```

-rw-r--r-- 1 root root 0 Feb 27 11:33 first  
-rw-r--r-- 1 root mail 0 Feb 27 11:40 second



## The Sticky Bit

- Normally users with write permissions to a directory can delete any file in that directory regardless of that file's permissions or ownership
- With the sticky bit set on a directory, only the owner of a file can delete the file
- Example:

```
ls -ld /tmp
drwxrwxrwt 12 root  root  4096 Nov 2 15:44 /tmp
```

4-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

It may seem counterintuitive that write permissions on a directory would allow one user to delete another user's file within that directory. Consider, however, that a directory is really just a file itself whose contents are references to other files. Deleting a file is therefore an edit to a directory file's list of other files.

Many users need to be able to create and delete files in /tmp. Even if users do not actively create files in /tmp, many applications they run will use /tmp as a location for temporary files. Setting the sticky bit prevents users from deleting each others' files, even though they have full access to the directory.

Note that the sticky bit on /tmp is set by default, and can be seen as a "t" in the file permissions:

```
drwxrwxrwt 13 root  root  4096 Sep 29 12:42 /tmp
```

To set the sticky bit on a directory, use **chmod**:

```
[root@stationX]# chmod o+t /home/share
```

Alternately:

```
[root@stationX]# chmod 1777 /home/share
```

The sticky bit will appear as a **T** if the directory's execute permission for "others" is off.

# Configuring the Quota System

- Overview
  - Implemented within the kernel
  - Enabled on a per-filesystem basis
  - Individual policies for groups or users
    - Limit by the number of blocks or inodes
    - Implement both soft and hard limits
- Initialization
  - Partition mount options: `usrquota`, `grpquota`
  - Initialize database: `quotacheck -cugm /filesystem/`

4-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Linux quota system allows an administrator to establish limits on the amount of disk resources users can consume. Because resource accounting must occur with every file creation, quotas are implemented within the kernel. Various required quota administration utilities are found within the quota RPM.

## Quota initialization

The following examples show how to implement user quotas on the `/home` partition. Group quotas are implemented in a nearly identical manner.

- Define partition options:

For a partition to implement quotas, it must be mounted with the `usrquota` or `grpquota` options. These options can be added to the appropriate entries in `/etc/fstab`. After you edit the file, the options can be made to immediately take effect by remounting the filesystem.

Edit `/etc/fstab`, adding the `usrquota` option to the `/home` partition:

```
# mount -o remount /home
```

- Create/Update the database:

The disk usage database is stored within a partition's top-level directory in specially named binary files, `aquota.user` and `aquota.group`. Use `quotacheck -cug` to create a new user and group quota file. During initialization, or any time the database file is out of sync with the actual state of a partition (for example, if quotas were turned off for a period of time, or if a system was brought down ungracefully, without unmounting partitions), the database can be brought up to date by running the `quotacheck` command:

```
# quotacheck -c /home
```

# Setting Quotas for Users

- Implementation
  - Start or stop quotas: **quotaon**, **quotaoff**
  - Edit quotas directly: **edquota username**
  - From a shell:

```
setquota username 4096 5120 40 50 /foo
```

- Define prototypical users:

```
edquota -p user1 user2
```

4-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (819) 754 3700.

- Starting and stopping quotas:

Quotas are turned on and off with the **quotaon** and **quotaoff** commands. These commands either take partitions as their arguments, or are invoked with the **-a** command line switch, in which case quotas are turned on for all appropriate partitions defined in `/etc/fstab`. These commands rarely need to be run in practice, because they are included within the default Red Hat Enterprise Linux initialization script `/etc/rc.d/rc.sysinit`.

```
# quotaon /home
```

```
# quotaon -a /home
```

- Editing user policies:

User policies are implemented with the **edquota** command. This command invokes an editor and loads a template, which can then be edited to establish the appropriate values. These values are committed to the database when you exit the editor. To ease the propagation of quotas, user policies can be prototyped from established policies for another user. Often, when first establishing quotas, it is helpful to first define a prototypical user, and then edit the user's properties. (If you do not first prototype the user, no template is provided by the **edquota** command.) Grace periods are established by using the **-t** command line switch with **edquota**.

```
# edquota user1
```

(implement a policy for *user1*)

```
# edquota -p user1 user2
```

(mimic the policy for *user2* from the policy for *user1*)

```
# edquota -t
```

(establish a grace period)

# Reporting Quota Status

- Reporting
  - User inspection: **quota**
  - Quota overviews: **repquota**
  - Miscellaneous utilities: **warnquota**

4-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

- Generating quota reports:

Users can inspect their disk usage and quotas by issuing the **quota** command. An administrator can generate a report of disk usages by all users with the **repquota** command. Users over their quotas can be warned with a **warnquota** cron job.

## End of Unit 4

- Questions and Answers
- Summary
  - userdel, chage
  - chmod g+s
  - chmod o+t
  - Filesystem quotas

4-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 4

## User and Group Administration

---

**Goal:** Understand user and group administration in Red Hat Enterprise Linux.

**System Setup:** A system with /home on a separate partition.

```
# mount | grep /home  
/dev/mapper/vol0-home on /home type ext3 (rw)
```

This should display a filesystem mounted on /home.

## Sequence 1: Creating the groups and users

**Scenario:** You need to set up groups for different departments in your company. You also need to set up user accounts for the employees in those departments.

**Deliverable:** A system with users joshua and alex in the sales group; dax and bryan in the hr group; zak and ed in the web group, and manager in the sales, hr, and web groups.

### Instructions:

1. Make certain that all newly created users will create group-writable files.
2. Add accounts for the following seven users to your system: joshua, alex, dax, bryan, zak, ed, and manager. Assign each user this password: password
3. Add the following groups to the system:
  - sales (GID:200)
  - hr (GID: 201)
  - web (GID: 202)
4. Why should you set the GID in this manner instead of allowing the system to set the GID by default?  
The presuable numerical group users
5. Add joshua and alex to the sales auxiliary group, dax and bryan to the hr auxiliary group. Add zak and ed to the web auxiliary group. Add manager to all of these auxiliary groups.
6. How can you add the web group to dax's auxiliary groups?  
usermod -G web dax
7. You can login as each user and use the **id** command to verify that they are in the appropriate groups. How else might you verify this information?  

---



## Sequence 2: Setting up shared directories

**Scenario:** Each department for which you created a group also needs a shared directory. This will allow users in each department to share files, but will prevent users in other departments from altering, or even seeing those files.

**Deliverable:** A shared directory for each department that allows only users in that department to enter it or create, view, and alter files within.

### Instructions:

1. Create a directory called `/depts` with `sales`, `hr`, and `web` subdirectories.
2. Set the group ownership of each directory to the group with the matching name.
3. Check the permissions on the `/depts` directory to verify that everyone can access, but not write to the directory. Change the permissions on each each subdirectory to grant full permissions (`rwX`) to the group and deny access to others.
4. Files created within those directories should be owned by the appropriate group. Set the appropriate permissions.
5. Experiment by logging in as each user and creating or altering files in each of the directories. Only manager should be able to enter all the directories.

## Sequence 3: Implementing Quotas

**Deliverable:** A user diskhog that cannot use more than 1024k of space in /home.

**Instructions:**

1. Create a user diskhog
2. Activate user quotas for /home
3. Set the soft block quota of user diskhog to 512 1k blocks and the hard limit to 1024 1k blocks.
4. Test the restrictions.

## Sequence 1 Solutions

1. Make certain that all newly created users will create group-writable files.
  - a. Add a umask entry to /etc/skel/.bashrc making group-writable files (002).

```
[root@stationX]# echo 'umask 002' >> /etc/skel/.bashrc
```

2. Add accounts for the following seven users to your system: joshua, alex, dax, bryan, zak, ed, and manager.

Since you need to add several users, a **for**-loop may speed things up. This can be entered on the command line or placed in a file and run as a shell script.

```
for USER in joshua alex dax bryan zak ed manager
do
    useradd $USER
    echo password | passwd --stdin $USER
done
```

Note that this sets a password of password for each user.

3. Add the following groups to the system:

- sales (GID:200)
- hr (GID: 201)
- web (GID: 202)

- a. [root@stationX]# **groupadd -g 200 sales**
- b. [root@stationX]# **groupadd -g 201 hr**
- c. [root@stationX]# **groupadd -g 202 web**

4. Why should you set the GID in this manner instead of allowing the system to set the GID by default?

The primary group for new users will be created with lowest available GID above 500. Most administrators want to keep additional groups in a specific range of GIDs to ease management.

5. Add joshua and alex to the sales auxiliary group, dax and bryan to the hr auxiliary group. Add zak and ed to the web auxiliary group. Add manager to all auxiliary groups.

You can use **usermod -G** to do this:

```
[root@stationX]# usermod -G sales joshua
[root@stationX]# usermod -G sales alex
[root@stationX]# usermod -G hr dax
[root@stationX]# usermod -G hr bryan
[root@stationX]# usermod -G web zak
```

```
[root@stationX]# usermod -G web ed
[root@stationX]# usermod -G sales,hr,web manager
```

6. How can you add the web group to dax's auxiliary groups?

```
[root@stationX]# usermod -G web,hr dax
```

-OR-

```
[root@stationX]# usermod -a -G web dax
```

If you had run the command **usermod -G web dax**, that would have removed dax from the hr group and any other auxiliary groups to which dax may have belonged.

7. You can login as each user and use the **id** command to verify that they are in the appropriate groups. How else might you verify this information?

You can use **su - user** and run the **id** command, or simply use the username as an argument to **id**:

```
for USER in joshua alex dax bryan zak ed manager
do
id $USER
done
```

## Sequence 2 Solutions

1. Create a directory called /depts with sales, hr, and web subdirectories.

```
[root@stationX]# mkdir -p /depts/{sales,hr,web}
```

2. Set the group ownership of each directory to the group with the matching name.

```
[root@stationX]# for GROUP in sales hr web
do
chgrp $GROUP /depts/$GROUP
done
```

3. Check the permissions on the /depts directory to verify that everyone can access, but not write to the directory. Change the permissions on each each subdirectory to grant full permissions (rwx) to the group and deny access to others.

```
a. [root@stationX]# ls -ld /depts
drwxr-xr-x 5 root root 4096 May  4 06:04 /depts
```

```
b. [root@stationX]# chmod g=rwx,o= /depts/*
```

-OR-

```
[root@stationX]# chmod 770 /depts/*
```

4. Files created within those directories should be owned by the appropriate group. Set the appropriate permissions.

```
[root@stationX]# chmod g+s /depts/*
```

5. Experiment by logging in as each user and creating or altering files in each of the directories. Only manager should be able to enter all the directories.

The easiest way is probably **su -**, but make sure to include the dash and to exit one **su** session before starting another.

```
[root@stationX]# su - joshua
[joshua@stationX]$ touch /depts/sales/test
[joshua@stationX]$ exit
[root@stationX]# su - alex
[alex@stationX]$ touch /depts/sales/test
[alex@stationX]$ exit
[root@stationX]# su - dax
[dax@stationX]$ touch /depts/hr/test
[dax@stationX]$ exit
[root@stationX]# su - bryan
[bryan@stationX]$ touch /depts/hr/test
[bryan@stationX]$ exit
[root@stationX]# su - zak
[zak@stationX]$ touch /depts/web/test
[zak@stationX]$ exit
```

```
[root@stationX]# su - ed
[ed@stationX]$ touch /depts/web/test
[ed@stationX]$ exit
[root@stationX]# su - manager
[manager@stationX]$ touch /depts/sales/test
[manager@stationX]$ touch /depts/hr/test
[manager@stationX]$ touch /depts/web/test
[manager@stationX]$ exit
```

## Sequence 3 Solutions

1. Create a user diskhog

```
# useradd diskhog
```

2. Activate user quotas for /home

- a. In `/etc/fstab` change the mount options for /home to include `usrquota`. If you do not have a separate /home partition use `.`

- b. Activate the new mount option.

```
# mount -o remount /home
```

- c. Bring the system into single user mode, to ensure correct quota calculation:

```
# init 1
```

- d. Run `quotacheck -cu /home` to create the quota file.

- e. Leave single user mode

```
# init 5
```

- f. Enable Quota enforcing.

```
# quotaon /home
```

3. Set the soft block quota of user diskhog to 512 1k blocks and the hard limit to 1024 1k blocks.

```
# setquota -u diskhog 512 1024 0 0 /home
```

4. Test the restrictions.

To test these restrictions, run the following commands:

```
su - diskhog
quota
dd if=/dev/zero of=bigfile bs=1k count=400
quota
```

(Should work fine.)

```
dd if=/dev/zero of=bigfile bs=1k count=800
quota
```

(Should issue a warning.)

```
dd if=/dev/zero of=bigfile bs=1k count=1600
```

quota

(Should fail to write the whole file.)



# Unit 5

## Local Security

5-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Configure the default Firewall
- Use Access Control Lists (ACLs)
- Manage SELinux
- Understand sudo
- Configure system logging

5-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Default Firewall

- Customized kernel level firewall
- All outbound connections are allowed
- **system-config-securitylevel**
  - *Trusted services, Other ports*
- Principally designed for workstations
- Does not co-exist well with manually entered **iptables** rules
- Can not be configured for additional *Trusted Services*

5-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Access Control List (ACL)

- Linux does not allow users to **chown** files
- ACLs allow users to share files without the risks of **chmod 777**
- Implemented as a mount option
  - **mount -o acl /directory**
- Set on the filesystem at install time
  - **tune2fs -l /dev/sda1 | grep options**

5-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The ext3 filesystem includes support for access control lists which allow finer grained control of file system permissions than are possible with the standard three access categories that are normally provided. Many filesystem commands, such as **cp** and **mv**, have been modified to copy the associated ACLs for a file.

In order to enable ACLs on a file system, the file system must be mounted with the **acl mount** option. To remount the **/home/** directory with the **acl** option, run the following:

```
[root@stationX]# mount -o remount,acl /home/
```

With Red Hat Enterprise Linux 5, **acl** is set as a default mount option by the installer. Rather than placing an **acl** entry in **/etc/fstab** for each partition, the installer sets the option in the ext3 header. The ext3 header can be viewed with either the **dumpe2fs** or **tune2fs -l**. The ACL option is set in the *Default mount options* field.

```
[root@stationX ~]# tune2fs -l /dev/sda1 | grep options
Default mount options:    user_xattr acl
```

When options are set on the filesystem level, however, they are not visible with **mount**.

```
[root@stationX ~]# mount | grep /dev/sda1
/dev/sda1 on /boot type ext3 (rw)
```

Filesystems created post-install will not have ACLs active by default. This feature can be set with **tune2fs -o**.

```
[root@stationX ~]# tune2fs -l /dev/sda5 | grep options
Default mount options:    (none)
[root@stationX ~]# tune2fs /dev/sda5 -o acl
tune2fs 1.39 (29-May-2006)
[root@stationX ~]# tune2fs -l /dev/sda5 | grep options
Default mount options:    acl
```

# ACL Usage

- Grant rwx access to files and directories for multiple users or groups
  - **getfacl** *file|directory*
  - **setfacl -m** *u:gandalf:rwx file|directory*
  - **setfacl -m** *g:nazgul:rw file|directory*
  - **setfacl -m** *d:u:frodo:rw directory*
  - **setfacl -x** *u:samwise file|directory*

5-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To view the ACLs for a file, use the **getfacl** command:

```
[root@stationX]# getfacl /home/schedule.txt
getfacl: Removing leading '/' from absolute path names
# file: home/schedule.txt
# owner: root
# group: root
user::rw-
user:bob:rwx
group::r--
group:admins:rw
mask::rwx
other::r--
```

ACLs can be modified using the **setfacl** command:

```
[root@stationX]# setfacl -m u:visitor:rx /home/schedule.txt
```

The above command would grant the user visitor read and execute access to the file /home/schedule.txt.

To remove (expunge) an ACL:

```
[root@stationX]# setfacl -x u:visitor /home/schedule.txt
```

On directories, default access control lists can be used. If we wanted all newly created content of a directory to be readable and writable by the user student:

```
[root@stationX]# setfacl -m d:u:student:rw /home/share/project/
```

# ACL Inheritance

- New files inherit default ACL (if set) from directory
- The `-p` option in the `cp` command preserves ACLs
- The `mv` command preserves ACLs

5-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

New files (either copied into a directory or newly created within) will inherit the default ACL of the directory (assuming that a default has been set). A default ACL for a directory can be set using the following command:

```
[root@stationX]# setfacl -m d:u:student:rx somedir
```

The above command will set a default ACL allowing user student to have read and execute access to all files created in directory `somedir`.

The `-p` option when used with the `cp` command will preserve ACLs, as well as any permissions which have been set on a file. This will not include any default ACLs.

The `mv` command will preserve any ACLs along with permissions when moving a file. Again, this will not include any default ACLs.

# SELinux

- Mandatory Access Control (MAC) -vs- Discretionary Access Control (DAC)
- A rule set called the *policy* determines how strict the control
- Processes are either restricted or unconfined
- The policy defines which resources a restricted process is allowed to access
- Any action that is not explicitly allowed is, by default, denied

5-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In an effort to deal with ever increasing threats to their data systems, the US government assigned the National Security Agency (NSA) the task of developing a single set of rules that all other agencies would follow in handling confidential information. By evaluating previous breaches, the NSA determined that a major hurdle to securing data was internal users bypassing local security. In some cases, users would inadvertently open access to a system. A classic example would be a user executing the command **chmod 777 ~**. To the user, this may seem an easy way to allow co-workers to share files. They may not realize that this gives everyone in the world access to potentially confidential information.

In more extreme cases, users would intentionally disable security for more insidious reasons. Both of these situations are examples of a user having the discretion to control the access to their system. The NSA felt the solution was to have systems implement *Mandatory Access Control* (MAC) over the users. In MAC, a set of rules, known as the *policy*, identify what a process is allowed to do. Anything that is not explicitly permitted is, by default, denied. Ideally, different policies could be implemented depending upon how strict the security needs.

The NSA's first implementation of MAC was a system called Mach, which introduced the concept of *Type Enforcement* (TE). Objects (files, directories, resources) were assigned a type value. Users and processes were also assigned a type value. The policy contains rules that allow a user or process to access objects. It was possible for a policy to be so *strict*, it was difficult to manage, so occasionally users and processes were allowed to be *unconfined*. This meant the policy did not apply to that user or process.

The NSA decided Mach made a better security framework than operating system. To get this framework deployed on production systems, the NSA needed access to the OS source code. They took advantage of the open source Linux kernel, and developed a set of patches to enable MAC. These patches became known as Security Enhanced Linux, or SELinux for short. They were later refined and incorporated into the kernel source by Red Hat.

## SELinux, continued

- All files and processes have a *security context*
- The context has several elements, depending on the security needs
  - user:role:type:sensitivity:category
  - user\_u:object\_r:tmp\_t:s0:c0
  - Not all systems will display s0:c0
- **ls -Z**
- **ps -Z**
  - Usually paired with other options, such as **-e**

5-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

There is an old UNIX saying that “everything is a file”. Traditional file access is controlled by user, group, and permission settings. To SELinux, everything is an object and access is controlled by security elements stored in the inode's extended attribute fields. Collectively, the elements are called the security context. Presently, there are five supported elements, although all five may not be present on all systems:

user	Indicates the type of user that is logged into the system. If a user logs in as root, they will have the user value of <code>root</code> . Other users will have a value of <code>user_u</code> . If they escalate their privileges with <code>su</code> , they will still have the user value of <code>user_u</code> . Processes have a value of <code>system_u</code> .
role	Defines the purpose of the particular file, process, or user. Files have the role of <code>object_r</code> . Processes get the role of <code>system_r</code> . Users also have the role of <code>system_r</code> , because (to Linux) users are similar to processes.
type	Used by Type Enforcement to specify the nature of the data in a file or processes. Rules within the policy say what process types can access which file types.
sensitivity	A security classification sometimes used by government agencies.
category	Similar to group, but can block root's access to confidential data.

To view the security context of a file, use the `ls` command's **-Z** option:

```
[root@shadex4 ~]# ls -Z /root/anaconda-ks.cfg /var/log/messages
-rw-r--r-- root root user_u:object_r:user_home_t anaconda-ks.cfg
-rw----- root root system_u:object_r:var_log_t /var/log/messages
```

Generally speaking, files inherit a directory's security context:

```
[root@stationX ~]# ls -Zd /etc /etc/hosts
drwxr-xr-x root root system_u:object_r:etc_t /etc
-rw-r--r-- root root system_u:object_r:etc_t /etc/hosts
```

Some files, however, get a unique security context, for added security:

```
[root@stationX ~]# ls -Z /etc/shadow /etc/aliases
```



```
-r----- root root system_u:object_r:shadow_t /etc/shadow
rw-r--r-- root root system_u:object_r:etc_aliases_t /etc/aliases
```

If a system were running under the most secure configuration, everything would be restricted by SELinux. This is not practical in most cases. For this reason, SELinux is being deployed in phases. In Red Hat Enterprise Linux 4, SELinux was protecting 13 processes. In the initial release of Red Hat Enterprise Linux 5, this count had increased to 88.

To determine if a process is protected, use to **ps** command's **-Z** option:

```
[root@stationX ~]# ps -ZC syslogd,bash
LABEL                                PID  TTY    TIME    CMD
system_u:system_r:syslogd_t         1888  ?      00:00:00 syslogd
user_u:system_r:unconfined_t        2583  pts/0  00:00:00 bash
```

Any process whose type is *unconfined\_t*, is not yet restricted by SELinux. To view the entire process stack, use either **ps -eZ** or **ps Zax**.

*Note:* A restricted process is sometimes called *protected*, though it is the data that is protected, not the process.

# SELinux: Targeted Policy

- The targeted policy is loaded at install time
- Most local processes are *unconfined*
- Principally uses the type element for *type enforcement*
- The security context can be changed with **chcon**
  - **chcon -t tmp\_t /etc/hosts**
- Safer to use **restorecon**
  - **restorecon /etc/hosts**

5-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The source code that allows SELinux to protect a process is integrated into the kernel, but the rules that define how SELinux enforces security is defined in the policy. The NSA's most secure policy is called the *strict* policy. By default, Red Hat uses the *targeted* policy. The targeted policy "targets" certain processes to be restricted, and then enforces their access to files and resources. Other policies also exist, such as the *Multi Level Security* (MLS). All policies could be thought of as a subset of the Strict policy.

The policy defines the elements that can be used, and whether users are allowed to manipulate the elements. Since the targeted policy uses Type Enforcement as its principal security mechanism, it pays the most attention to the type element of the security context. The policy allows the **chcon** command to change the security context.

```
[root@stationX ~]# ls -Z install.log
-rw-r--r-- root root root:object_r:user_home_t install.log
[root@stationX ~]# chcon -t etc_t install.log
[root@stationX ~]# ls -Z install.log
-rw-r--r-- root root root:object_r:etc_t install.log
```

The **chcon** command can only use types that are defined in the policy. Rather than memorizing all the possible types that can be used, **chcon --reference** can take the security context from one object, and apply it to another.

```
[root@stationX ~]# ls -Z anaconda-ks.cfg
-rw----- root root system_u:object_r:user_home_t anaconda-ks.cfg
[root@stationX ~]# chcon --reference /etc/shadow anaconda-ks.cfg
[root@stationX ~]# ls -Z anaconda-ks.cfg
-rw----- root root system_u:object_r:shadow_t anaconda-ks.cfg
```

A safer alternative is the **restorecon** command. With **restorecon**, the policy determines and applies the object's default context.

```
[root@stationX ~]# restorecon /root/*
[root@stationX ~]# ls -Z /root
```

-rw----- root root root:object\_r:user\_home\_t anaconda-ks.cfg  
-rw-r--r-- root root root:object\_r:user\_home\_t install.log

# SELinux: Management

- Modes: Enforcing, Permissive, Disabled
  - Changing enforcement is allowed in the Targeted policy
  - **getenforce**
  - **setenforce 0 | 1**
  - Disable from GRUB with **selinux=0**
- **/etc/sysconfig/selinux**
- **system-config-securitylevel**
  - Change mode, Disabling requires reboot
- **system-config-selinux**
  - Booleans
- **setroubleshootd**
  - Advises on how to avoid errors, not ensure security!

5-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The policy for a system is defined in the `/etc/sysconfig/selinux` file as is the mode of operation. The policy can be Disabled, Enforcing, or Permissive. Disabled means the policy is ignored. Permissive is a mode for troubleshooting or development that logs policy violations, but does not prevent programs from running. Enforcing is the default mode.

The **getenforce** command can be used determine the system's current mode. The targeted policy allows the use of the **setenforce** command to toggle between the Enforcing (1) and Permissive (0) mode:

```
[root@stationX ~]# getenforce
Enforcing
[root@stationX ~]# setenforce 0
[root@stationX ~]# getenforce
Permissive
```

The only way to disable SELinux is to change `/etc/sysconfig/selinux`, and reboot, or use the **selinux=0** Grub kernel option.

The GUI tool, **system-config-selinux**, allows for changes of a few other SELinux options. The targeted policy allows certain features to be controlled through a set of *booleans*. These are predefined functions that can selectively turn off enforcement of certain daemons. It would be preferable to ensure objects have the correct security context rather than shutting off security features.

When an application attempts something that is not authorized by the policy, SELinux blocks access and an error is logged to `/var/log/audit/audit.log` in case auditd is running (this is the default). When auditd is not running, SELinux logs to `/var/log/messages`. The application is often unaware

of why it failed. This can make troubleshooting difficult. To help in the troubleshooting process, the **setroubleshootd** daemon will alert you of the error by placing a warning icon on the alert panel. Clicking on the icon will display a possible fix for the error. It is important to realize that the proposed solution may not be the best solution for the problem.

## SUID and SGID Executables

- Normally processes started by a user run under the user and group identity of that user
- SUID and/or SGID bits set on an executable file cause it to run under the user and/or group identity of the file's owner and/or group

5-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When a user starts a process, it runs with the permissions of that user. For example, if you run **vi**, and try to edit a file which you do not have permission to read or write, the operation will fail. However, if the SUID or SGID bit is set on an executable, it runs with the permissions of its owner (or group owner). For example, consider the file `/etc/shadow` that stores users' encrypted passwords:

```
-r----- 1 root root 805 Sep 29 11:19 /etc/shadow
```

The file is owned by root, who has exclusive read access. Users may still change their passwords with the **passwd** command, because the **passwd** command has its SUID bit set, and is owned by root:

```
-rwsr-xr-x 1 root root 13536 Jul 12 05:56 /usr/bin/passwd
```

SUID and SGID bits are set using the **chmod** command:

SUID:

```
[root@stationX]# chmod u+s filename
```

SGID:

```
[root@stationX]# chmod g+s filename
```

Since the SUID and SGID permissions are displayed overlying the execute permission for either user or group, respectively, the case of the permission indicates whether the execute permission is turned on or off. For example, if a capital is in the execute field, the SUID or SGID permission is on and the execute permission is off. If a lowercase is in the execute field, both the SUID or SGID and the execute permissions are on.

For security reasons, SUID and SGID permissions are not honored when set on non-compiled programs, such as shell scripts.

# sudo

- Users listed in `/etc/sudoers` execute commands with:
  - an effective user id of 0
  - group id of root's group
- An administrator will be contacted if a user not listed in `/etc/sudoers` attempts to use **sudo**

5-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The purpose of **sudo** is to delegate root privileges to non-root users. It has many advantages over the **su** including not having to manage a shared password and logging of who actually executed commands, when and from where.

**sudo** access is controlled by `/etc/sudoers`. `/etc/sudoers` contains mappings of variables to reference groups of users, hosts, or commands. This file should be edited with **visudo**, which will open a copy of the current file with **vi** (or whatever editor is specified in the `EDITOR` environment variable) and check the syntax before implementing changes.

One common `sudoers` configuration is to grant full privileges to a user or group of trusted users with the expectation that they'll use the privileges responsibly and honestly. This simple configuration can be established with just one line in the `sudoers` file:

```
username ALL=(ALL) ALL
```

To give a specific group of users limited root privileges, edit the file with **visudo** as follows:

In the user alias specification section, list users and groups allowed to use the **sudo** command:

```
User_Alias LIMITEDTRUST=student1,student2
```

In the command alias specification section, list the commands specifically allowed or denied execution:

```
Cmdnd_Alias MINIMUM=/etc/rc.d/init.d/httpd
```

In the user privilege specification section, list the users and groups allowed to use **sudo** and the commands that they may use:

```
LIMITEDTRUST ALL=MINIMUM
```

Users `student1` and `student2` can use **sudo** only with the commands listed with **MINIMUM (httpd)**.

When delegating privileges to users that are not completely trusted, extreme caution must be employed. Many commands commonly used by root were not written with this sort of security in mind and are not appropriate for **sudo** access. Any command that launches an editor, pager or can be configured through environment variables is probably not safe. Shell scripts are also generally not safe unless they take care to reset the `PATH` environment variable.

# System Logging

- Centralized logging daemons: **syslogd**, **klogd**, **auditd**
- Log file examples:
  - `/var/log/dmesg`: Kernel boot messages
  - `/var/log/messages`:#Standard system error messages
  - `/var/log/secure`: Security, authentication, and xinetd messages
- **syslogd** Configuration:
  - `/etc/sysconfig/syslog`
  - `/etc/syslog.conf`
- **auditd** Configuration:
  - `/etc/audit/auditd.conf`
  - `/etc/audit/audit.rules`
- Application log files and directories also reside in `/var/log`

5-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A variety of log files are maintained by the system, an understanding of which is often vital for troubleshooting system problems:

`/var/log/dmesg` -- This log file is written upon system boot. It contains messages from the kernel that were raised during the boot process.

`/var/log/messages` -- This is the standard system log file, which contains messages from all your system software, non-kernel boot issues, and messages that go to `dmesg`. Readable only by root.

`/var/log/secure` -- This log file contains messages and errors from security-related systems such as `login`, `tcp_wrappers`, and `xinetd`. Readable only by root. Very useful in detecting and investigating network abuse.

`/var/log/maillog` -- This log file contains messages and errors from your sendmail. Readable only by root.

There are also various other system log files that store information from other applications (i.e. Apache, Squid, etc.) that also may be found under `/var/log`.



## End of Unit 5

- Questions and Answers
- Summary
  - system-config-securitylevel
  - ACLs
  - SELinux Management
  - sudo
  - syslog

5-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 5

## Local Security

---

**Goal:** Understand system security features effecting local operations.

**System Setup:** A system with /opt mounted to a dedicated partition.

```
# mount | grep /opt  
/dev/sda5 on /opt type ext3 (rw)
```

**Situation:** A new system has been installed in your network that will be accessed locally and remotely by several users

## Sequence 1: Working with ACLs

**Scenario:** An organization's three accountants, Amy, Arthur, and Ann, need the ability to control access to a set of files. None of the accountants should have root access, but each must be able to authorize write access as needed. The files should be available, but not by the organization's director, Austin.

**Deliverable:** A set of filesystems supporting ACLs.

### Instructions:

1. Add an `acct` group. Add four user accounts for Amy, Arthur, Ann, and Austin, all in the `acct` group. The users need not have passwords for this exercise. Verify that root can `su -` to Amy, with the home directory of `/home/amy`. As root, create a `/acct` as a directory in the root filesystem (this is the default behavior.) The directory should be owned by Amy, assigned to the `acct` group, with permissions of 755.

For the purposes of this exercise, set the permissions for `/opt` to 1777.

2. As Amy, use ACLs to grant read, write, and execute access to the `/acct` directory to Arthur and Ann. Investigate write privileges. Switch to the Arthur, Ann, and Austin, accounts and have each try to create a file in the directory. Austin's should fail.
3. As Amy, create two files in `/acct`: `cayman` and `swiss`. Allow Arthur to write to `cayman` and Ann to write to `swiss`. Create a subdirectory called `banks`. Set the ACL such that all files created in `banks` are writable by Amy, Arthur, and Ann, but not Austin.
4. As Amy, create `/acct/banks/deposits`. Move the file `deposits` file to `/tmp`. Investigate the ACLs for the file. Move the file to `/opt`. Investigate the ACLs for the file.
5. For some reason, the ACLs were not preserved during the move between `/tmp` and `/opt`. Remember that ACLs are a mount option. Execute `mount` and review the partitions and options. You may notice that none of the partitions are showing the `acl` as an option.  
  
If a partition is created at install time, the `acl` option is set in the filesystem's header. We can view these headers with `tune2fs -l`. Since `/acct` is not a mount point, it is on the root filesystem. Examine the root filesystem's header, paying special attention to the *Default mount options*. Compare it with the `/opt` filesystem's header.
6. To fix the problem, we need to ensure that the system boots with `/opt` mounted with the `acl` option. Use `/etc/fstab` to specify that rather than accepting the defaults for `/opt`, you want to use ACLs. Execute `mount /opt -o remount` to test your changes.

## Sequence 2: Understanding file context.

**Scenario:** While restoring files from a backup, it has come to your attention that the certain daemons can not access the files. Review the security context of the files and the processes to understand the way SELinux restricts access.

**Deliverable:** A properly working SELinux security sub-system.

### Instructions:

1. Ensure SELinux is in the *Enforcing* mode.
2. Launch a window to monitor errors written to `/var/log/audit/audit.log` as follows:  

```
# xterm -e "tail -f /var/log/audit/audit.log" &
```
3. View the security context of the **syslogd** process. View the processes configuration file, `/etc/syslog.conf`. Notice that it references several log files for various resources, including **cron**, which is written to `/var/log/cron`.
4. Use **service syslog stop** to stop the process. Copy `/var/log/cron` to root's home directory. View the new file's security context. Move `/var/log/cron` to `/tmp`. View the file's security context. What do you notice about the behavior of copy and move?
5. Move the **cron** file that is in root's home directory to `/var/log`. View its security context. Start the **syslog** service. Within a few moments an error should appear in the monitoring window indicating an *avc: denied* condition.
6. There are several ways to fix this problem. One is to restore the original from `/tmp`. Another is to use **chcon** to assign an alternate security context and hope we get the right one. A better choice might be to see if **restorecon** knows the correct type. Attempt a **restorecon** and restart the service.

## Sequence 3: SELinux Booleans

**Scenario:** SELinux is restricting a single daemon, but all others are functioning properly. Rather than disabling SELinux for the entire system, a boolean value will be used to disable protection of the individual service.

**Deliverable:** A service running with the *initrc\_t* type.

### Instructions:

1. Investigate the security context for the **syslogd** process. Remember that this is different from the security context for the file.
2. We want to use **system-config-selinux**, but it may not be installed. Use your **yum** skills to determine the package required. Install the package.
3. Launch **system-config-selinux**. Select the *Boolean* selection. Scroll down to the *SELinux Service Protection* item. Expand the selection by clicking the triangle. Continue to scroll down to the *Disable SELinux protection for syslogd daemon*. Check the box to disable the capacity for SELinux to restrict **syslogd**. This may take a few seconds as it makes this change permanent. Exit the application.
4. Restart the **syslogd** daemon and investigate its security context.
5. Re-enable the boolean and restart the service.

## Sequence 1 Solutions

1. Add an acct group. Add four user accounts for Amy, Arthur, Ann, and Austin, all in the acct group. The users need not have passwords for this exercise.

```
# groupadd acct
# for J in amy arthur ann austin; do useradd $J -G acct; done
# ls /home
amy arthur ann austin student visitor
# su - amy
$ pwd
/home/amy
```

As root, create a /acct as a directory in the root filesystem (this is the default behavior.) The directory should be owned by Amy, assigned to the acct group, with permissions of 755.

```
# mkdir /acct
# chown amy /acct; chgrp acct /acct
# ls -ld /acct
drwxr-xr-x 2 amy acct 4096 Mar  1 18:41 /acct
```

For the purposes of this exercise, set the permissions for /opt to 1777.

```
# chmod 1777 /opt
```

2. As Amy, use ACLs to grant read, write, and execute access to the /acct directory to Arthur and Ann.

```
# su - amy
$ setfacl -m u:arthur:rwX /acct
$ setfacl -m u:ann:rwX /acct
```

View the results.

```
$ ls -ld /acct/
drwxrwxr-x+ 2 amy acct 4096 Mar  1 21:33 /acct/
$ getfacl /acct/
getfacl: Removing leading '/' from absolute path names
# file: acct
# owner: amy
# group: acct
user::rwX
user:arthur:rwX
user:ann:rwX
group::r-x
mask::rwX
other::r-x
```

We could have made the directory writable by the group with **chmod g+w**, but that would have given austin permission to create files. Yet, looking at the listing of /acct, it seems as if it is writable by the group. Notice the plus symbol. With ACLs, it is the **getfacl** information

that is relevant, not the standard permissions. In this case, the system is trying to tell us that the directory is writable by more than one user.

Switch to the Arthur, Ann, and Austin, accounts and have each try to create a file in the directory. Austin's should fail.

```
# su - arthur
$ touch /acct/arthur-file; exit
# su - ann
$ touch /acct/ann-file; exit
# su - austin
$ groups
austin acct
$ ls -ld /acct/
drwxrwxr-x+ 2 amy acct 4096 Mar  1 21:33 /acct/
$ ls /acct/
-rw-rw-r-- 2 ann    ann    0 Mar  1 21:35 ann-file
-rw-rw-r-- 2 arthur arthur 0 Mar  1 21:37 arthur-file
$ touch /acct/testfile
touch: cannot touch `/acct/testfile': Permission denied
```

Look back at the output of **getfacl**. Notice that the group only has r-x permission.

3. As Amy, create two files in /acct: cayman and swiss. Allow Arthur to write to cayman and Ann to write to swiss.

```
$ touch cayman swiss
$ setfacl -m u:arthur:w cayman
$ setfacl -m u:ann:w swiss
$ ls -l
-rw-rw-r--+ 1 amy amy 0 Mar  1 21:49 cayman
-rw-rw-r--+ 1 amy amy 0 Mar  1 21:49 swiss
```

Use **getfacl** to verify the permissions.

Create a subdirectory called banks. Set the ACL such that all files created in banks are writable by Amy, Arthur, and Ann, but not Austin. Create the file /acct/banks/deposits and investigate which users can write to the file.

```
$ mkdir /acct/banks
$ ls -ld /acct/banks/
drwxrwxr-x 2 amy amy 4096 Mar  1 21:52 /acct/banks/
$ setfacl -m d:u:arthur:rw banks
$ setfacl -m d:u:ann:rw banks
$ getfacl /acct/banks/
# file: banks
# owner: amy
# group: amy
user::rwx
group::rwx
```

```
other::r-x
default:user::rw
default:user:arthur:rw
default:user:ann:rw
default:group::rwx
default:mask::rwx
default:other::r-x
```

Notice how Amy used the "d" in the ACL. This means the settings will be the default for file creation.

Create the file /acct/banks/deposits and investigate which users can write to the file.

```
$ touch /acct/banks/deposits
$ getfacl /acct/banks/deposits
... output truncated ...
user::rw-
user:arthur:rw-
user:ann:rw-
... output truncated ...
```

4. As Amy, Move the file deposits file to /tmp. Investigate the ACLs for the file.

```
$ getfacl /acct/banks/deposits
... output truncated ...
$ mv /acct/banks/deposits /tmp
$ getfacl /tmp/deposits
... output truncated ...
```

The information should be identical: the ACLs were preserved with the move.

Move the file to /opt. Investigate the ACLs for the file.

```
$ mv /tmp/deposits /opt
mv: preserving permissions for `/opt/deposits': Operation not supported
mv: preserving ACL for `/opt/deposits': Operation not supported
mv: preserving permissions for `/opt/deposits': Operation not supported
```

Oops. That doesn't look good.

5. For some reason, the ACLs were not preserved during the move between /tmp and /opt. Remember that ACLs are a mount option. Execute **mount** and review the partitions and options. You may notice that none of the partitions are showing the **acl** as an option.

If a partition is created at install time, the **acl** option is set in the filesystems header. We can view these headers with **tune2fs -l**. Since /acct is not a mount point, it is on the root filesystem. Examine the root filesystems header.



```
# mount | grep "/"  
/dev/sda2 on / type ext3 (rw)  
# tune2fs -l /dev/sda2 | grep acl  
Default mount options: user_xattr acl
```

Notice that `acl` appears in *Default mount options*. Check the default mount options for `/opt`.

```
# mount | grep "/opt"  
/dev/sda5 on /opt type ext3 (rw)  
# tune2fs -l /dev/sda5 | grep "Default mount options"  
Default mount options: (none)
```

At issue is the fact that the installer created the root filesystem, but `/opt` was created manually, post install.

6. To fix the problem, we need to ensure that the system boots with `/opt` mounted with the `acl` option. Use `/etc/fstab` to specify that rather than accepting the defaults for `/opt`, you want to use ACLs.

```
# grep "/opt" /etc/fstab  
LABEL=/opt          /opt          ext3      defaults    0 0  
# vi /etc/fstab  
# grep "/home" /etc/fstab  
LABEL=/opt          /opt          ext3      acl         0 0
```

Execute **`mount /opt -o remount`** to test your changes.

```
# mount | grep "/opt"  
/dev/sda5 on /opt type ext3 (rw)  
# mount /opt -o remount  
# mount | grep "/opt"  
/dev/sda5 on /opt type ext3 (rw,acl)
```

## Sequence 2 Solutions

1. Ensure SELinux is in the *Enforcing* mode.

```
# setenforce 1
```

2. Launch a window to monitor errors written to `/var/log/audit/audit.log` as follows:

```
# xterm -e "tail -f /var/log/audit/audit.log" &
```

3. View the security context of the **syslogd** process. View its configuration file, `/etc/syslog.conf`, and notice that it references several log files for various resources. Messages from **cron** would be written to `/var/log/cron`. View the security context of the log file.

```
# ps -eZ | grep syslog
root:system_r:syslogd_t          9094 ?                00:00:00 syslogd
# grep "^cron" /etc/syslog.conf
cron.*                          /var/log/cron
# ls -Z /var/log/cron
-rw----- root root system_u:object_r:var_log_t cron
```

4. Use **service syslog stop** to stop the process.

```
# service syslog stop
Shutting down kernel logger:    [ OK ]
Shutting down system logger:    [ OK ]
```

Copy `/var/log/cron` to root's home directory. View the new file's security context.

```
# cp /var/log/cron ~
# ls -Z ~/cron
-rw----- root root root:object_r:user_home_t cron
```

Move the original `/var/log/cron` to `/tmp`. View the file's security context.

```
# mv /var/log/cron /tmp
# ls -Z /tmp/cron
-rw----- root root system_u:object_r:var_log_t cron
```

Notice that as files are moved around the system, the security context follows the file. If a file is copied, the original file keeps its security context, but the new copy inherits its context from its parent directory.

5. Move the `cron` file that is in root's home directory to `/var/log`. View its security context.

```
# mv /root/cron /var/log
# ls -Z cron
-rw----- root root root:object_r:user_home_t cron
# service syslog start
Starting system logger:         [ OK ]
Starting kernel logger:        [ OK ]
```

Start the **syslog** service. Within a few moments an error should appear in the monitoring window indicating an *avc: denied* condition.

```
type=AVC msg=audit(1174474933.386:260): avc: denied
{ append } for pid=13224 comm="syslogd" name="cron" dev=dm-0
ino=1900555 scontext=user_u:system_r:syslogd_t:s0
tcontext=user_u:object_r:user_home_t:s0 tclass=file
```

Notice that message indicates that SELinux is preventing the *command* **syslogd** from accessing the file with the *name* of *cron*.

6. There are several ways to fix this problem. One is to restore the original from /tmp. Another is to use **chcon** to assign an alternate security context and hope we get the right one. A better choice might be to see if **restorecon** knows the correct type. Attempt a **restorecon** and restart the service.

```
# ls -Z /var/log/cron
-rw----- root root root:object_r:user_home_t      cron
# restorecon /var/log/cron
# ls -Z /var/log/cron
-rw----- root root system_u:object_r:var_log_t     cron
# service syslog restart
Shutting down kernel logger:      [ OK ]
Shutting down system logger:      [ OK ]
Starting system logger:           [ OK ]
Starting kernel logger:           [ OK ]
```

Continue to monitor the logs for a few moments to be sure there are no additional messages.

## Sequence 3 Solutions

1. Investigate the security context for the **syslogd** process. Remember that this is different from the security context for the file.

```
[root@stationX]# ps -eZ | grep syslogd
system_u:system_r:syslogd_t 8760 ? 00:00:00 syslogd
```

Notice that the type element is `syslogd_t`.

2. We want to use **system-config-selinux**, but it may not be installed. Use your **yum** skills to determine the package required. Install the package.

```
a. [root@stationX]# yum whatprovides system-config-selinux
policycoreutils-gui.i386
Matched from:
...
```

```
b. [root@stationX]# yum -y install policycoreutils-gui
```

3. Launch **system-config-selinux**. Select the *Boolean* selection. Scroll down to the *SELinux Service Protection* item. Expand the selection by clicking the triangle. Continue to scroll down to the *Disable SELinux protection for syslogd daemon*. Check the box to disable the capacity for SELinux to restrict **syslogd**. This may take a few seconds as it makes this change permanent. Exit the application.

4. Restart the **syslogd** daemon and investigate its security context.

```
[root@stationX]# service syslog restart
Shutting down kernel logger: [ OK ]
Shutting down system logger: [ OK ]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
[root@stationX]# ps -eZ | grep syslogd
root:system_r:initrc_t 9536 ? 00:00:00 syslogd
```

The `initrd_t` is a special type that prevents SELinux from restricting the daemon.

5. Re-enable the boolean and restart the service.
  - a. Run through **system-config-selinux** and de-select **Disable SELinux protection for syslogd daemon**. Exit the application.
  - b. [root@stationX]# **service syslog restart**

# Unit 6

## Advanced Partitioning

6-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Setup and manage software Raid devices
- Configure Logical Volumes

6-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# What is Software RAID?

- Multiple disks grouped together into "arrays" to provide better performance, redundancy or both.
- **mdadm** - provides the administration interface to software RAID.
- Many "RAID Levels" supported, including RAID 0, 1, 5 and 6.
- Spare disks add extra redundancy
- RAID devices are named, `/dev/md0`, `/dev/md1`, `/dev/md2`, `/dev/md3` and so on.

6-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Either partitions or whole disks can be used to create software RAID devices. The term "disk" is used here to denote either a whole disk or a partition one or more disks. Though it's possible to create an array composed entirely of partitions on a single device, as is demonstrated in this class, there's little benefit in doing so outside of the lab environment.

The most commonly used RAID types are:

- **RAID 0 or Striping:** Two or more disks used to create a single large high performance volume. Performance is better if drives of equal size are used. No redundancy, so chance of failure is very high. Array size equals the sum of all disks in array.
- **RAID 1 or Mirroring:** Two disks containing the the same data updated simultaneously. Redundancy offers good protection against disk failure. Can slow write performance but tends to improve read performance. Only RAID type that you can place the `/boot` partition on. Hot spare disks can be used to improve fault-tolerance. Array size equals the size of the smallest disk used.
- **RAID 5:** Three or more disks with zero or more hot spares. A good balance between performance and reliability. Redundancy is achieved by splitting parity between all disks, one disk can be lost without causing array failure. Both read and write speeds are usually improved, but in certain cases write performance is dramatically decreased. For this reason RAID 5 is often not a good choice to host databases.
- **RAID6, or striping with dual (duplicated) distributed parity,** is similar to RAID5 except that it improves fault tolerance by allowing the failure of any two drives in the array. While the simultaneous failure of two devices may be an unlikely event, RAID6 protects the array from data loss during recovery of a single disk failure, provides the administrator additional time to perform rebuilds, improves the viability of less expensive drives in enterprise storage solutions, and, since RAID only passively checks for bad blocks, provides protection from undiscovered block errors.

Once an array is created it's device, `/dev/md0` for example, is used the same way as `/dev/hda6` or `/dev/sda6` may have been used in an earlier example.

# Software RAID Configuration

- Create and define RAID devices using **mdadm**

```
mdadm -C /dev/md0 -a yes -l 1 -n 2 -x 1 elements...
```

- Format each RAID device with a filesystem

```
mke2fs -j /dev/md0
```

- Test the RAID devices
- **mdadm** allows you to check the status of your RAID devices

```
mdadm --detail /dev/md0
```

6-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To install and configure software RAID after installation, the **mdadm** RPM is required. It should already be installed by default, since it is included in the minimal Core installer component.

1. Use **fdisk** or another tool to create disk partitions of type 0xfd, Linux RAID.
2. If necessary, run **partprobe** to have the kernel reload the partition table.
3. Initialize and activate your RAID array:

```
[root@stationX]# mdadm -C /dev/md0 --chunk=64 --level=5 --raid-devices=3  
/dev/sd{b,c,d}1
```

--level or -l sets the RAID level. --raid-devices or -n sets the number of RAID disks.  
--spare-device or -x optionally sets the number of hot spares. -a yes creates the device file if they do not exist.

4. Create a filesystem on the new software RAID device. With **mke2fs**, there is a special **-R stride=n** option that can improve performance. The stride is the software RAID device's chunk-size in filesystem blocks. For example, with an ext3 filesystem that will have a 4 KB block size on a RAID device with a chunk-size of 64 KB, the stride should be set to 16:

```
[root@stationX]# mke2fs -j -b 4096 -R stride=16 /dev/md0
```

5. Add the software RAID device and its mount point to **/etc/fstab** as appropriate.



# Software RAID Testing and Recovery

- Simulating disk failures

```
mdadm /dev/md0 -f /dev/sda1
```

- Recovering from a software RAID disk failure

- replace the failed hard drive and power on
- reconstruct partitions on the replacement drive
- `mdadm /dev/md0 -a /dev/sda1`

- `mdadm`, `/proc/mdstat`, and syslog messages

6-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `mdadm -f` command can be used to simulate a drive failure. This is useful when testing RAID 1 or RAID 5 arrays, but will destroy a RAID 0 array. Spare disks can be set aside in a RAID 1 or RAID 5 array, that can automatically start rebuilding as a replacement disk ( `-x` option ). The array can be used while it is rebuilding, although performance will be degraded.

Recovery is fairly simple. Power off the system, replace the failed disk, power on the system, and reconstruct an appropriate partition table on the disk. The new disk can be inserted into the array with the `mdadm -a` command. You may have to remove the failed disk from the array with `mdadm -r`.

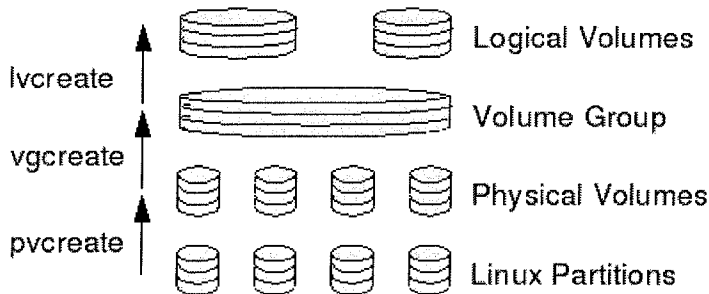
Information about disk failures is logged through syslog to `/var/log/messages` by default. Information on the current state of software RAID devices is also available in `/proc/mdstat`. The output below shows a RAID 5 device, `/dev/md0`, made up of `/dev/sdb1`, `/dev/sdc1`, and `/dev/sdd1`; `/dev/sdd1` has failed.

```
[root@stationX ~]# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 sdd1[3] (F) sdc1[1] sdb1[0]
272896 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU_]
unused devices: <none>
```

Information on estimated time to reconstruction will appear in this file if the array is rebuilding. You can also use the `mdadm --detail` command to view the status of the RAID array.

# What is Logical Volume Manager (LVM)?

- A layer of abstraction that allows easy manipulation of volumes. Including resizing of filesystems
- Allows reorganization of filesystems across multiple physical devices
  - Devices are designated as Physical Volumes
  - One or more Physical Volumes are used to create a Volume Group
  - Volume Groups are defined with Physical Extents of a fixed size
  - Logical Volumes are created on Volume Groups and are composed of Physical Extents
  - Filesystems may be created on Logical Volumes



6-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

LVM creates a higher-level layer of abstraction than traditional linux disks and partitions. This allows for great flexibility in allocating storage. Logical volumes can be resized and moved between physical devices easily. Physical devices can be added and removed with relative ease. LVM managed volumes can also have sensible names like "database" or "home" rather than the somewhat cryptic "sda" or "hda" device names.

# Creating Logical Volumes

- Create physical volumes

```
pvcreate /dev/hda3
```

- Assign physical volumes to volume groups

```
vgcreate vg0 /dev/hda3
```

- Create logical volumes from volume groups

```
lvcreate -L 256M -n data vg0  
mke2fs -j /dev/vg0/data
```

6-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Any disk partitions to be used as physical volumes need to have their partition types set to 0x8e, Linux LVM. Then the partitions need to be added as physical volumes with `pvcreate` device-name. Software RAID devices may also be set up as physical volumes.

New volume groups need to be created and one or more physical volumes assigned to them with the `vgcreate` command. The new volume group can have almost any name. At this time the size of an extent is determined; by default it is 4 MB. This affects the minimum size of changes which can be made to a logical volume in the volume group, and the maximum size of logical and physical volumes in the volume group. A logical volume can contain at most 65534 extents, so the default extent size limits the volume to about 256 GB; a size of 1 TB would require extents of at least 16 MB:

```
vgcreate -s 16M vg0 /dev/hda3 /dev/sda2
```

Logical volumes are created from these volume groups, and may have arbitrary names. The size of the new volume may be requested in either extents or in KB, MB, GB, or TB (rounding up to whole extents):

```
lvcreate -L 512M -n data vg0
```

(asks for /dev/vg0/data of 512 MB)

```
lvcreate -l 32 -n test vg0
```

(asks for /dev/vg0/test of 32 extents)

Logical volumes may also be striped like a RAID 0 device between multiple physical volumes. A striped logical volume may be extended later, but only with extents from the original physical volumes. Because LVM can not tell if two particular physical volumes in the volume group are on the same drive, it is best not to do this if a volume group will contain striped logical volumes. The following command creates a logical volume, `/dev/vg1/striped`, striped between two physical volumes in the `vg1` volume group, using the default stripe size.

lvcreate -i 2 -L 1G -n striped vg1

# Resizing Logical Volumes

- Growing Volumes
  - **lvextend** can grow logical volumes
  - **resize2fs** can grow EXT3 filesystems online
  - **vgextend** adds new physical volumes to an existing volume group.
- Shrinking volumes
  - Filesystem must be reduced first
  - Requires a filesystem check and cannot be performed online
  - **lvreduce** can then reduce the volume.
- Volume Groups can be reduced with:

```
pvmove /dev/hda3  
vgreduce vg0 /dev/hda3
```

6-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Logical volumes can be resized dynamically while preserving the data on the volume, if the volume's filesystem supports resizing. The **lvextend** command allows resizing of an e or ext3-based logical volume. **resize2fs** can be used to grow mounted ext2 and ext3 filesystems. **lvextend** must be called first to grow the logical volume.

The following commands grow the mounted /dev/vg0/data filesystem:

```
# lvextend -L +500M /dev/vg0/data  
# resize2fs /dev/vg0/data
```

For other filesystems, the **lvextend** utility can be used to add unallocated extents in the volume group to a logical volume. Then native utilities for the filesystem can be used to expand it to fill the volume. To reduce a filesystem, first use the native utilities to shrink the filesystem, then run **lvreduce** to shrink the logical volume.

```
# df /data  
Filesystem              1K-blocks      Used Available Use% Mounted on  
/dev/mapper/vg0-data    253871      87846    152918   37% /data  
# umount /data  
# e2fsck -f /dev/vg0/data  
e2fsck 1.39 (29-May-2006)  
Pass 1: Checking inodes, blocks, and sizes  
Pass 2: Checking directory structure  
Pass 3: Checking directory connectivity  
Pass 3A: Optimizing directories  
Pass 4: Checking reference counts
```

## Pass 5: Checking group summary information

```
/dev/vg0/data: ***** FILE SYSTEM WAS MODIFIED *****
```

```
/dev/vg0/data: 1908/65536 files (0.8% non-contiguous), 96122/262144 blocks
```

```
# resize2fs /dev/vg0/data 128M
```

```
resize2fs 1.39 (29-May-2006)
```

```
Resizing the filesystem on /dev/vg0/data to 131072 (1k) blocks.
```

```
The filesystem on /dev/vg0/data is now 131072 blocks long.
```

```
# lvreduce -L 128M /dev/vg0/data
```

```
WARNING: Reducing active logical volume to 128.00 MB
```

```
THIS MAY DESTROY YOUR DATA (filesystem etc.)
```

```
Do you really want to reduce data? [y/n]: y
```

```
Reducing logical volume data to 128.00 MB
```

```
Logical volume data successfully resized
```

```
# mount /dev/vg0/data /data
```

```
# df /data
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/mapper/vg0-data	126931	87337	34352	72%	/data

Additional physical volumes can be added to a volume group to provide more unallocated extents to assign to logical volumes. The physical volumes need to be setup with **pvccreate**, then added to the volume group with the **vgextend** command.

Physical volumes can also be removed from a volume group. This is useful for removing an old disk from the volume group. The **pvmove** command can redistribute extents from the physical volume being decommissioned to the other physical volumes in the volume group. In its simplest mode, **pvmove** takes the name of the physical volume to be removed as its argument; for example:

```
# pvmove /dev/hda3
```

Once there are no extents in use on the old physical volume, it can be removed from the volume group with the **vgreduce** command:

```
# vgreduce vg0 /dev/hda3
```

Some commands are available to help you gather information about the state of your physical volumes, volume groups, and logical volumes. Three useful commands are **pvdiskdisplay**, **vgdisplay**, and **lvdisplay**.

## End of Unit 6

- Questions and Answers
- Summary
  - Configuration of Software RAID
  - Software RAID recovery
  - Configuration of Logical Volumes

6-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[ttraining@redhat.com](mailto:ttraining@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 6

## Advanced Partitioning

---

**Goal:** Develop skills and knowledge related to Software RAID and LVM.

**Lab Setup:** The LVM subsystem caches metadata in `/etc/lvm/.cache`. You may need to delete this file in order to see your changes.

```
# rm /etc/lvm/.cache
# vgscan
```

If you continue to see AVC errors when you run LVM commands, you may need to fix the SELinux context of the LVM cache file.

```
# restorecon /etc/lvm/.cache
```



## Sequence 1: Working With Software RAID

**Scenario:** A mission critical application needs to be installed on a redundant media device. The decision has been made to use software RAID on an existing system. Modify the configuration of your system to support a mirrored array with a hot spare device.

**Deliverable:** A system with a RAID 1 with auto-fail-over.

### Instructions:

1. Create four 100Mb partitions with type of `fd`. Use the first three partitions to create a mirrored RAID device with one spare. Create an `ext3` filesystem on the array and mount it to `/raid`.
2. View the RAID's status using `--detail` and `/proc/mdstat`.
3. Execute the command: `xterm -e "watch cat /proc/mdstat" &`  
  
This will launch a separate window for monitoring your array. Fail the second element of the array with the `-f` option and observe the effects in the monitoring window. Keep the monitoring window open during this exercise.
4. Add the fourth partition to the array. View with `--detail`. What is the fourth elements status? What is the status of the other three? Why didn't the array return the third element to the spare status?
5. The second element still shows as failed. Remove it from the array, and observe any changes in the status of the other three elements.

## Sequence 2: Creating A Logical Volume

**Scenario:** As a method of sharing the new RAID device with multiple applications, deploy LVM on the RAID device. One logical volume will need to be created with an ext3 filesystem.

**Deliverable:** A resizable volume on a redundant device.

### Instructions:

1. Unmount your RAID from the previous exercise. Use the mirrored device as a physical volume for a volume group named alpha. Set the extent size to 32M.
2. Display the physical volume. Why is some of the space unusable? Display the volume group. How many extents are available?
3. Build a 50M logical volume with the name of beta and an ext3 filesystem.
4. Edit your `/etc/fstab` to include a line that will mount `/dev/alpha/beta` to the `/gamma` mount point. Execute **mount -a** to ensure the device will mount properly.

## Sequence 3: Extending A Logical Volume

**Scenario:** An new volume group is needed for a service that requires the frequent addition of space. Create the group with two partitions such that new partitions could be added, and the filesystem could be extended.

**Deliverable:** A filesystem whose size can easily be changed.

**Instructions:**

1. Determine which RAID partition is not currently in use by `/dev/md0`. Use **fdisk** to change the partition's type from `fd` to `8e`. Create two additional partitions with sizes of 256M and 312M, with a partition type of `8e`.
2. Use the 100M partition and the 256M partition as physical volumes for a group called *delta* with an extent size of 4M.
3. Create a logical volume of 200M named *epsilon* with an `ext3` filesystem. Add an entry to `fstab` to mount the volume at boot time to the mount point of *zeta*. Use **mount -a** to test.
4. Copy the contents of `/usr` to `/zeta`. Use **df** to confirm this has filled logical volume. Display the volume group to determine the number of free extents.
5. Extend the volume group by adding the last partition. Extend the logical volume to 400M. Extend the filesystem.

## Sequence 4: Reduce a Logical Volume

**Scenario:** An older application's data needs are not growing as originally projected. Unused space needs to be reallocated. Reduce to filesystem, returning the extents to the group.

**Deliverable:** Free extents available for other logical partitions.

### Instructions:

1. Use **df** to determine the amount of free space available in /zeta. We will reduce it by 50M.
2. Install and launch **system-config-lvm**. Once the display loads, you will see several horizontal cylinders. Highlight one of the segments and review the status display in the right hand pane.
3. In the left hand pane, expand the **Logical View** item. Highlight the epsilon item. In the middle pane, click the button labeled **Edit Properties**. A new window will open with the logical volumes specifications. Change the **LV size** setting from Extents to Megabytes. Reduce the size of the logical volume by 50M. If it reads 350M, reduce it to 300M. Click **OK**.
4. The application will warn that the logical volume is mounted and must be unmounted to continue. Confirm the warning by clicking **Yes**. Once complete, exit the application, and confirm the resizing with **df**.

If time allows, explore the application in more detail.

## Sequence 1 Solutions

1. Create four 100Mb partitions with type of fd.

```
# fdisk /dev/sda
```

Ensure you use the `t` directive to set the type values before saving your changes. When you are finished, your partitioning scheme may look similar to this:

```
# partprobe
# fdisk -l
... output truncated ...
/dev/sda7      222      233      1023860      fd      Linux      Raid      Auto
/dev/sda8      234      245      1023860      fd      Linux      Raid      Auto
/dev/sda9      245      256      1023860      fd      Linux      Raid      Auto
/dev/sda10     256      267      1023860      fd      Linux      Raid      Auto
```

Use the first three partitions to create a mirrored RAID device with one spare.

```
# mdadm -C /dev/md0 -l 1 -n 2 -x 1 /dev/sda7 /dev/sda8 /dev/sda9
```

Create an ext3 filesystem on the array and mount it to `/raid`.

```
# mke2fs -j /dev/md0
# mkdir /raid
# mount /dev/md0 /raid
```

2. View the RAID's status with `--detail` and `/proc/mdstat`.

```
# mdadm --detail /dev/md0
... output truncated ...
/dev/sda7      active
/dev/sda8      active
/dev/sda9      spare
# cat /proc/mdstat
Personalities: raid1
/dev/md0      /dev/sda7 /dev/sda8 /dev/sda9
```

3. Execute the command: `xterm -e "watch cat /proc/mdstat" &`

This will launch a separate window for monitoring your array. Fail the second element of the array with the `-f` option and notice the effects in the monitoring window.

```
# mdadm /dev/md0 -f /dev/sda8
```

You should see a progress bar indicating the status of activating the spare disk. When recovery is complete, the failed partition will be tagged with (F).

Keep the monitoring window open during this exercise.

4. Add the fourth partition to the array.

```
# mdadm /dev/md0 -a /dev/sda10
hot add: /dev/sda10
```

View with **--detail**. What is the fourth element's status? What is the status of the other three? Why didn't the array return the third element to the spare status?

```
# mdadm --detail /dev/md0
... output truncated ...
/dev/sda7    active
/dev/sda8    failed
/dev/sda9    active
/dev/sda10   spare
```

The new element has been added as a spare. This does not change the status of the other elements. The array does not care which element is the spare, so the status will only be changed as a result of a real or simulated failure.

5. The second element still shows as failed. Remove it from the array, and observe any changes in the status of the other three elements.

```
# mdadm /dev/md0 -r /dev/sda8
hot remove: /dev/sda8
```

Again, the array's status will remain unchanged.

## Sequence 2 Solutions

1. Unmount your RAID from the previous exercise.

```
# umount /dev/md0
```

Use the mirrored device as a physical volume for a volume group named alpha.

```
# pvcreate /dev/md0
# vgcreate alpha -s 32M /dev/md0
```

2. Display the physical volume. Why is some of the space unusable?

```
# pvdisplay /dev/md0
--- Physical volume ---
PV Name                /dev/md0
VG Name                alpha
PV Size                100 MB / not usable 4 MB
... output truncated ...
```

The unusable space is a result of having chosen an inefficient combination of options. Since 100 is not evenly divisible by 32, we have wasted space. Given the 100M device, we should have chosen 4M to avoid this situation.

Display the volume group. How many extents are available?

```
# vgdisplay alpha
--- Volume group ---
VG Name                alpha
... output truncated ...
VG Size                100.00 MB
PE Size                32.00 MB
Total PE               3
Alloc PE / Size        0 / 0 MB
Free PE / Size         3 / 96 MB
```

There are a three physical extents (PE) available, totalling 96M of disk space.

3. Build a 50M logical volume with the name of beta. Display the volume's status, then create an ext3 filesystem on the device.

```
# lvcreate -L 50M alpha -n beta
Rounding to full physical extent
Created 64M logical volume
# lvdisplay
--- Logical volume ---
LV Name                /dev/alpha/beta
VG Name                alpha
... output truncated ...
# mke2fs -j /dev/alpha/beta
```

... output truncated ...

4. Edit your `/etc/fstab` to include a line that will mount `/dev/alpha/beta` to the `/gamma` mount point. The new entry may look similar to the following:

```
/dev/alpha/beta      /gamma      ext3    defaults    0 0
```

Execute **mount -a** to ensure the device will mount properly. (Did you remember to create the mount point?)

```
# mount -a
# mount | grep beta
/dev/mapper/alpha-beta on /gamma type ext3 (rw)
```

Notice that the kernel does not refer to the device with the exactly as we typed the name.



## Sequence 3 Solutions

1. Determine which RAID partition is not currently in use by `/dev/md0`. This could be accomplished by comparing an **fdisk -l** with `/proc/mdstat`. In the following example, partitions 7, 9, and 10 are in use. This would mean 8 is available.

```
# fdisk -l | grep fd
Disk /dev/md0 doesn't contain a valid partition table
/dev/sda7      889    901    104391    fd    Linux raid autodetect
/dev/sda8      902    914    104391    fd    Linux raid autodetect
/dev/sda9      915    927    104391    fd    Linux raid autodetect
/dev/sda10     928    940    104391    fd    Linux raid autodetect
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 hda10[2] hda9[1] hda7[0]
      104320 blocks [2/2] [UU]
```

Use **fdisk** to change the partitions type from `fd` to `8e`. Create two additional partitions with sizes of 256M and 312M, with a partition type of `8e`. Your partition table may look similar to this example:

```
/dev/sda7      889    901    104391    fd    Linux raid autodetect
/dev/sda8      902    914    104391    8e    Linux LVM
/dev/sda9      915    927    104391    fd    Linux raid autodetect
/dev/sda10     928    940    104391    fd    Linux raid autodetect
/dev/sda11     941    972    257008+   8e    Linux LVM
/dev/sda12     973    1011   313236    8e    Linux LVM
```

Commit your changes to disk. Remember that you must **partprobe** after changing the partition table.

2. Use the 100M partition and the 256M partition as physical volumes for a group called *delta* with an extent size of 4M.

```
# pvcreate /dev/sda8 /dev/sda11
Physical volume "/dev/sda8" successfully created
Physical volume "/dev/sda11" successfully created
# vgcreate delta -s 4M /dev/sda8 /dev/sda11
Volume group "delta" successfully created
```

3. Create a logical volume of 200M named *epsilon* with an ext3 filesystem. Use **mount -a** to test.

```
# lvcreate delta -L 200M -n epsilon
Logical volume "epsilon" created
# mke2fs -j /dev/delta/epsilon
... output truncated ...
# mkdir /zeta
```

Add an entry to `fstab` to mount the volume at boot time to the mount point of *zeta*.

```
/dev/delta/epsilon    /zeta                ext3    defaults    0 0
```

Use **mount -a** to test. Troubleshoot as needed.

```
# mount -a
# mount | grep epsilon
/dev/mapper/delta-epsilon on /zeta type ext3 (rw)
```

4. Copy the contents of /usr to /zeta. Use **df** to confirm this has filled logical volume. Display the volume group to determine the number of free extents.

```
# cp -r /usr /zeta/
# df | grep zeta
      198337      198337      0 100% /zeta
# vgdisplay delta
--- Volume group ---
VG Name                delta
... output truncated ...
Alloc PE / Size         50 / 200.00 MB
Free  PE / Size         37 / 148.00 MB
```

5. Extend the volume group by adding the last partition. Extend the logical volume to 400M. Extend the filesystem.

```
]# pvcreate /dev/sda12
   Physical volume "/dev/sda12" successfully created
# vgextend delta /dev/sda12
   Volume group "delta" successfully extended
# lvextend -L 400M /dev/delta/epsilon
   Extending logical volume epsilon to 400.00 MB
   Logical volume epsilon successfully resized
# df | grep zeta
      198337      198337      0 100% /zeta
# resize2fs /dev/delta/epsilon
resize2fs 1.35 (28-Feb-2004)
# df | grep zeta
      396672      199096      177111      53% /zeta
```

## Sequence 4 Solutions

1. Use **df** to determine the amount of free space is available in /zeta. We will reduce it by 50M.

```
# df | grep zeta
396672 199096 177111 53% /zeta
```

2. Install and launch **system-config-lvm**. Once the display loads, you will see several horizontal cylinders. Highlight one of the segments and review the status display in the right hand pane.

3. In the left hand pane, expand the **Logical View** item. Highlight the epsilon item. In the middle pane, click the button labeled **Edit Properties**. A new window will open with the logical volumes specifications. Change the **LV size** setting from Extents to Megabytes. Reduce the size of the logical volume by 50M. If it reads 350M, reduce it to 300M. Click **OK**.

4. The application will warn that the logical volume is mounted and must be unmounted to continue. Confirm the warning by clicking **Yes**. Once complete, exit the application, and confirm the resizing with **df**.

```
# df | grep zeta
346672 199096 226111 72% /zeta
```

If time allows, explore the application in more detail.

# Unit 7

## Installation

7-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Recall important command line switches
- Understand different installation methods
- Create advanced partition layouts
- Understand Kickstart's role
- Create Kickstart files

7-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Anaconda, the Red Hat Enterprise Linux Installer

- Supports different modes
  - Kickstart offers automated Installation
  - Upgrade performs an update of an existing Red Hat Enterprise Linux installation
  - Rescue Mode allows troubleshooting of unbootable systems
- Consists of two stages:
  - First stage starts the installation
  - Second stage performs the installation

7-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 526 2994 or +1 (919) 754 3700.

Anaconda is the installer for Red Hat Enterprise Linux. The Python based program features several modes. *Kickstart* mode provides an automated installation. This mode is discussed later in this unit.

The *Upgrade* mode can be selected when the Installer detects an existing Red Hat Enterprise Linux installation. It upgrades existing packages and leaves configuration and user data intact.

The *Rescue* mode can be used to troubleshoot a system that is no longer bootable. It loads the necessary rescue environment from the second stage and gives the user a shell to access the system. This mode is discussed in the Troubleshooting unit.

Anaconda is run in two stages. The first stage boots the system and performs initialization of the system. It will then load the second stage from the selected installation medium.

General information about the installation program are part of the **anaconda** RPM. If **anaconda** is installed, documentation can be found in `/usr/share/doc/anaconda-*/`. For additional information check the Red Hat Enterprise Linux Installation Guide on <http://www.redhat.com/docs>.

# First Stage: Starting the Installation

- The first stage consists of a installation kernel and an `initrd.img`
- Can be started with any supported bootloader
- Tasks of the First Stage:
  - Initializes the Installer
  - Parses command line arguments
  - Auto-detects hardware
  - Loads additional drivers
  - Selects language, keyboard layout and installation method
  - Sets up networking if required for installation

7-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The first stage loads Anaconda, the Red Hat Enterprise Linux installer. It consists of a special kernel and `initrd.img`. These files can be loaded by any supported bootloader. Images for alternative installation boot media can be found on the first installation disk or installation DVD.

The first stage parses command line arguments and then initializes the selected mode of the Installer. If booted from a installation CD it automatically starts the second stage. Otherwise it queries for language, keyboard layout and installation source. It sets up networking if the user selects a network based installation method. You can specify the **askmethod** command line argument to force a network installation from an installation CD.

Hardware is auto-detected by the installer. This may hang the installation under rare conditions. In this case specify **noprobe** to disable auto-detection.

Some hardware requires additional drivers to function. The hardware vendor can provide a driver disk to enable the device during installation. The installer automatically prompts for such a disk if auto-detection fails or the **dd** command line argument is used.

# First Stage: Boot Media

- Supported boot media:
  - `boot.iso` or Installation CD/DVD
  - USB drive containing `bootimg.img`
  - Network boot with PXE
  - Other bootloaders such as GRUB
  - Boot floppies no longer supported
- Boot media can be modified for custom installations

7-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Boot CDs

The `boot.iso` file contains a small CD filesystem image that can be burned to CD with the `cdrecord` command:

```
cdrecord -v boot.iso
```

You can create your own `boot.iso` to customize the installation process

1. Put the contents of the `boot.iso` image into a temporary directory.

```
# mkdir /tmp/iso
# mount -o loop boot.iso /mnt
# cd /mnt
# cp -a . /tmp/iso
```

2. Put additional files (e.g. kickstart files) into the same directory.
3. Edit `isolinux/isolinux.cfg` to modify/add boot entries.
4. You can also modify `isolinux/boot.msg` to provide a custom boot message.
5. Create a new bootable image with the following commands:

```
# cd /tmp/iso
# mkisofs -o ../bootcd.iso -b isolinux/isolinux.bin -c \
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -Jr .
```

## USB Drives

The `diskboot.img` file contains a VFAT filesystem image designed to be written to a USB storage device with the `dd` or `cat` commands. For example, on an already installed linux system, a USB



thumb-drive might be accessed as `/dev/sda1`. A boot image can be created on the USB device with the command:

```
cat diskboot.img > /dev/sda1
```

*(WARNING! This will also destroy any data that was on the device)*

Of course, this will only be useful if the BIOS supports booting from USB storage devices. Booting from a custom USB device may be preferred because it is the easiest way to customize the installation process.

## Network Boot

PXE, the Pre-eXecution Environment, provides a method of automating your install such that no physical media needs to be inserted into the target system (if the BIOS supports it). Information on configuring PXE can be found in Red Hat Enterprise Linux 4 System Administration Guide:

<http://www.redhat.com/docs/manuals/enterprise/>

# Accessing the Installer

- Graphical Installation
  - Default installation type
  - Useful Switches: `lowres`, `resolution`, `skipddc`
- VNC based Installation
  - Activate with **vnc** and protect the session with **vncpassword=password**
  - Set network parameters with **ip=IP Address** and **netmask=Network Mask**
- Text based Installation
  - Started with the **text** switch
  - Menu-based terminal interface
- Serial Installation
  - Used automatically when no graphic card is detected
  - Enable with: **serial=device**

7-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The graphical interface makes installation easy and intuitive. The graphical interface can be started in **lowres** mode, which means it uses lower screen resolution settings for the installation. The **resolution** switch can be used to specify a resolution such as 1024x768. If monitor detection does not work properly it can be disabled by using the **skipddc** switch.

The VNC interface is identical to the graphical interface. Once the second Stage of the installer is running you can access it with the **vncviewer**. Alternatively you can use the **vncconnect** switch, so that the installer connects to an listening vncviewer.

The text based installer is useful when the installer has difficulty managing your display adapter. While this is uncommon, it can be particularly useful on laptops that have proprietary display adapters. The text based installation does not support all installer features.

# First Stage: Installation Method

- Available Installation Methods:
  - Local CDROM
  - Hard drive
  - NFS image
  - FTP
  - HTTP
- Media sets:
  - Two Sets available: Client and Server
  - Can be downloaded from Red Hat Network
  - May contain packages from additional layered products
  - An "Installation Number" must be entered to unlock additional content
  - Extra packages can also be installed after installation through RHN.

7-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

## Installation Methods

Red Hat Enterprise Linux can be installed via several methods in addition to the traditional installation CDs. The user can place the ISO images of the media set onto EXT2 or VFAT filesystems on a partition of a hard disk, even external USB drives.

For NFS based installations the images or their contents must be located on a NFS share. FTP or HTTP installations require that the contents of the CDs are put into a public directory.

## Media Sets

Red Hat Enterprise Linux is now delivered on two distinct media sets: Client and Server. These media sets can be downloaded from RHN at [https://rhn.redhat.com/network/software/download\\_isos.pxt](https://rhn.redhat.com/network/software/download_isos.pxt) Media sets may also contain packages from certain layered products.

To avoid installation of packages which are not covered by the support contract, an Installation Number must be entered to enable these additional features. If the key is omitted only the base system is installed. The packages can also be obtained from RHN once the system is registered.

## Second Stage: Installation Overview

- Language and keyboard selection
- Installation Number
- Disk partitioning
- Bootloader configuration
- Network and time zone configuration
- Package selection

7-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The second stage Installer first configures language and keyboard mapping if this was not been done by the first stage. It then requests an Installation Number to access additional packages on the installation media. If the Installation Number is skipped, only packages from the base distribution will be installed.

Then Partitioning and bootloader configuration is performed. You can also specify a bootloader password. After the Network and Time Zone screens package selection allows customization of the installed RPMs.

# Configuring File Systems

- Must select mount points, partition sizes, and file system types in the installer
  - Can set up manually or automatically
- There are many layouts which may be used
  - / must include `/etc`, `/lib`, `/bin`, `/sbin`
  - Swap space is typically 2x physical RAM
  - Typical mount points: `/boot`, `/home`, `/usr`, `/var`, `/tmp`, `/usr/local`, `/opt`

7-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

## Configuring the file system hierarchy

At installation time, you must divide up your disks into partitions of various sizes and identify whether those partitions should be formatted with a file system (typically ext3), used as swap space, or used as a RAID or LVM partition. If the partition contains a file system, it must also be assigned a mount point. You can have the installer automatically make these decisions, or you can make them manually.

If you choose automatic configuration, you still have some input in the partitioning process. You can ask to review and modify the selections manually after the installer makes its decisions. You can select which drives to use for the installation. You can also indicate if the installer should delete all existing partitions, delete all Linux partitions from previous installations, or leave all existing partitions alone (using unallocated disk space for new partitions).

You have a great deal of freedom in how you may manually configure your file system hierarchy. You must have a file system mounted on `/`. You typically should have about twice your RAM in swap space. The `/etc` `/lib` `/bin` and `/sbin` directories may not be on separate file systems; they must be part of the `/` file system or the system will not boot properly.

It is common to have a `/boot` file system about 100 MB in size at the front of the disk, to hold files needed by the BIOS at boot time (such as the kernel and parts of the bootloader). This helps to avoid problems with old BIOS code. One limitation on `/boot` is that most bootloaders expect it to be on a normal disk partition or RAID 1 device.

The `/var` directory holds files that change frequently. This includes log files, the mail spool, and temporary space for software updates from Red Hat Network. If `/var` is on a separate partition, it should probably be at least 1 GB in size.

Depending on how much software you choose to install, `/usr` will probably need between 1 GB and 5 GB of space. The `/tmp` directory should have enough free space to accommodate temporary files written by programs. The amount of space needed will vary depending on the number and type of applications and services running on the system. One gigabyte should be more than enough for most systems' `/tmp` directories. The rest of `/` requires a few hundred megabytes. This does not take into account any space needed for personal user files or software not included with Red Hat Enterprise Linux.

# Advanced Partitioning

- **Software RAID**
  - Create new partitions and select **Software RAID** as “filesystem” type
  - Combine RAID partitions into a RAID device with **RAID**
- **LVM**
  - Select **Physical Volume** to create physical volumes
  - **LVM** creates a Volume Group
  - **Add** creates new Logical Volumes

7-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## RAID

To create a RAID device first create partitions with the RAID “filesystem” type. Then select **RAID** to configure RAID level and partitions to use. Make sure that partitions forming the RAID device are located on different physical disks.

Anaconda automatically configures `/etc/mdadm.conf` to allow monitoring of the RAID devices. After installation edit the file to select another email address for email notifications.

## LVM

A Volume Group is automatically created by the default partitioning. Select **LVM** to add more logical volumes. By default all available space is used. It might be advisable to leave some space unused in the Volume Group.

## Limitations

The System's BIOS must be able to access the second stage of the bootloader in `/boot`. `/boot` cannot be placed on a RAID device except on RAID1. It can not be part of a Volume Group. It is therefore recommended to put `/boot` on a separate filesystem.

## Package Selection

- A default set of packages is automatically installed
- Select **Customize now** to change the default set of packages
- Customizing is necessary to add support for additional languages
- Anaconda automatically resolves package dependencies
- Package set can easily customized after install with **yum** or **system-config-packages**

7-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda installs a default set of packages unless you select **Customize now** during installation. This is required to add support for additional languages. It is easy to customize the package set after installation using **yum** or **system-config-packages**. Therefore it is advised to only install a minimal package set during installation.

# First Boot: Post-Install Configuration

- Configure X Window System if necessary
- Firewall and SELinux Setup
- Kdump setup
- Set date and time
- Register with Red Hat Network and get updated RPMs
- Setup users
- Configure sound card
- Install additional RPMs or Red Hat documentation from CDROM

7-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux runs a graphical program called firstboot if the system is booted into run-level 5 after the installation. firstboot offers several configuration functions such as setting the date and time, installing additional software, or registering for Red Hat Network.

## Firewall Setup

The default firewall configuration blocks most incoming connection by default. You will be presented with two choices for the firewall, **Enabled** and **Disabled**.

**Trusted Services** allows you to let remote machines access particular services through the firewall. Some common services are listed. After the install is complete, you can use **system-config-securitylevel** to allow additional services in the **Other ports:** dialog box. This box takes a series of port:protocol pairs. For **port** you may use either the name from `/etc/services` or the port number; for example, both `imap` and `517` are acceptable definitions.

Firewall rules are written to `/etc/sysconfig/iptables` and invoked at boot up by the `/etc/rc.d/init.d/iptables` script.

## SELinux

SELinux is automatically put into Enforcing mode. You can change the mode and configure booleans in the **Modify SELinux Policy** tab during first boot.

## Kdump

Kdump can perform crash dumps. If a kernel panic occurs, the system switches to a second kernel image that then dumps key parts of the memory into swap space. For this some memory must be reserved.



# Kickstart

- Scripted Installation method
- Supports all Anaconda features
- `/root/anaconda-ks.cfg` is automatically created during Install
- Configuration utility: **system-config-kickstart**
- Syntax-checker: **ksvalidator**

7-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Kickstart allows the installer to read information from a designated file rather than prompting the person doing the installation for it. It especially facilitates the setup of a number of machines that have similar hardware that the installer can autoprobe successfully. If a required item is omitted from the Kickstart file, the installation pauses and the user is prompted for that information.

Anaconda automatically generates a Kickstart file during installation and saves it under `/root/anaconda-ks.cfg`. This file can be used as a basis for your own modifications.

**system-config-kickstart** is a graphical tool for creating and modifying Kickstart files. A separate syntax checker is also available: **ksvalidator**

# Starting a Kickstart Installation

- Anaconda enters Kickstart mode, when the `ks` boot option is specified
- `ks` queries DHCP for the Kickstart location
- `ks=url` gets the file via HTTP, FTP, or NFS
- From a local medium: `ks=floppy`, `ks=cdrom`, or `ks=hd:device:/path/to/file`

7-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Anaconda enters Kickstart mode when the `ks` option is given on the command line. Example:

```
boot: linux ks=http://server1/workstation.cfg
```

## DHCP based Kickstart

A DHCP server simplifies network-based Kickstarts, even if the system to be built will use a static IP address. It may also instruct the Kickstarted client about the location of the Kickstart file and the server and NFS share on which it may be found. If there is no DHCP available network configuration must be entered manually.

If the location given is a file, then the client tries to mount the file's parent directory to retrieve the file. If the location given is a directory, then the client tries to mount that directory and looks for a file whose name is *DHCP-IP-Address-Kickstart*.

If the DHCP server does not provide a `next-server` name, then the client assumes the DHCP is the server on which the Kickstart file resides. If the DHCP server does not provide a filename, then the client assumes the directory is */Kickstart* and looks for a Kickstart file with a name of the *DHCP-IP-Address-Kickstart* form.

## URL based Kickstart

It is also possible to specify a URL for the Kickstart file, like this:

```
linux ks=http://server/path/to/kickstart
```

This URL can point to a CGI script or other dynamic pages. If your Kickstart file is provided by NFS use this syntax:

```
linux ks=nfs:/server:file
```

## Kickstart from Local Media

Kickstart files can also be placed on local media.

<code>ks=floppy</code>	<code>ks.cfg</code> located in the top level directory of a EXT2 or VFAT floppy
------------------------	---

ks=cdrom	ks.cfg located in the top level directory of a CDROM
ks=hd:device:file	file is located on a VFAT or EXT2 filesystem.

# Anatomy of a Kickstart File

- **Commands section**
  - Configures the system
  - Omitted directives are prompted to the user
- **%packages Section**
  - Selects packages and groups for installation
  - Dependencies are always resolved
- **Scripts section(s)**
  - Optional section to customize the system
  - %pre scripts are run before installation
  - %post scripts are run after installation

7-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A Kickstart file consists of several sections. In the first section the system is configured. If one of the mandatory directives is omitted, Anaconda will prompt the user. After that it continues with Kickstart.

The %packages section determines which packages are installed. The base system will always be installed, even if this section is empty.

At the end of the Kickstart file you can have optional %pre and %post scripts. These are used to customize the installation.

# Kickstart: Commands Section

## Starting the Installation

- Installation Mode
  - `install` performs a fresh install.
  - `upgrade` upgrades an existing installation.

- Installation Method:

```
cdrom
url --url url
nfs --server host --path directory
harddrive --partition=device --dir=/path/to/install_tree
```

7-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Kickstart: Commands section

### Important Directives

- **Required Directives**
  - Must be specified, otherwise the installer configures them interactively
  - Localization options: `keyboard`, `lang`, `timezone`
  - Authentication: `rootpw`, `authconfig`
  - Bootloader: `bootloader`
- **Optional Directives**
  - Network: `network` [`options`]
  - Security: `firewall`, `selinux`, `services`
  - Installer behavior: `firstboot`, `poweroff` | `reboot`, `interactive`, `text`

7-17

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Required directives must be specified or the installer prompts the user. Optional directives are silently set to their defaults if omitted.

### Required Directives

The `keyboard` directive sets keyboard mapping (e.g. `keyboard de-latin1` or `keyboard us`). The language is set with the `lang` directive. It takes the same arguments as the `LANG` environment variable. If set to something except `en_US` make sure to install the appropriate `@lang-support` package groups. `timezone` specifies the time zone of the system. Use the `--utc` option if your hardware clock is set to GMT.

The root password can be either be specified in clear text (`rootpw secret`) or crypted (`rootpw --iscrypted $1$Y2Qix4sC$dZS9MTJ4iscVkoVy9hNGu/`). Use the `authconfig` directive to control authentication. If no parameters are given the system uses local authentication and `md5` password hashes.

The `bootloader` directive normally does not require arguments. Use `--password` or `--md5pass` to specify a bootloader password.

### Security

Firewall settings can be controlled with the `firewall` directive. The `--disabled` or `--enabled` options determine if `iptables` should be activated. It is possible to specify with incoming network connections are allowed with the options `--ssh`, `--telnet`, `--http`, `--smtp`. Other ports can be specified with the `--port` option. This option takes comma separated pairs of `port:protocol`.

Selinux can be put into `--permissive` or `--enforcing` mode or completely `--disabled`.

Services can automatically started after installation. The `services` directive has two arguments: `--disabled` and `--enabled`. Both take a comma separated list of services.

## Installer Behavior

The `firstboot` directive specifies, if **firstboot** should be executed after installation. it can be `--enabled` (the default) `--disabled`. There is also a special `--reconfig` option, that cause **firstboot** to reconfigure language, keyboard and timezone.

You can specify if the system should be rebooted (`reboot`) or powered down (`poweroff`) after installation. If not specified the system prompts for reboot.

Kickstart can be turned into a interactive installation with default values with the `interactive` directive. You can run the Kickstart also in text mode with the `text` keyword.

## Kickstart: Packages Section

- Add single packages with `package_name` without any version number
- Add package groups with `@package_group`
- Remove packages from the list: `-package_name`
- Use wildcards to specify multiple packages
- Dependencies are always resolved
- Add support for additional languages with `@lang-support`
- Packages from layered products can be installed when an installation number is specified by with the `key` directive in the commands section.

7-18

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `%packages` section adds packages to the installation. You can specify single packages, package groups (using `@package_group`) or remove packages included by a group with `-package_name`. Note that there is no version information, so the package section will also work with other releases with little changes.

### Package Groups

Package groups are defined in `/path/to/install-media/Install-Class/repodata`. `Install-Class` is `Server` or `Client` for the core product. If an installation number is specified additional install classes are available.

It is no longer possible to specify `@Everything` for an all-inclusive installation. If you really want to install all available packages, use the new wildcard feature:

```
%packages
```

```
*
```

Support for additional languages can be added by installing `language-name-support` package groups. e.g

```
%packages
```

```
@german-support
```



## Kickstart: %pre, %post

- %pre gives you the first word
  - executes as a bash shell script
  - executes after Kickstart file is parsed
- %post gives you the final word
  - Can specify interpreter (bash is default)
  - chrooted by default, but may be run without chroot

7-19

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The %pre and %post sections make just about anything possible. The %pre section executes as a bash script after the Kickstart file is read, but before partitioning, formatting, and copying of packages commences. The %pre script runs in a somewhat limited environment, as the only executables available are the ones provided by the installer. Unlike %pre, all of the packages specified in the %packages section are available in the %post section, which means many more utilities and capabilities are available.

The default behavior of %post is to execute the content of the section as a bash shell script in an environment that chroots to /mnt/sysimage -- the newly installed system. In other words, all paths and commands are as they will be on the installed system. The default use of bash as the interpreter may be overridden using the --interpreter switch on the %post line, e.g., %post --interpreter /usr/bin/perl. A non-chrooted environment is also possible through the --nochroot switch.

### Examples:

```
%pre
# Create partitions by copying an MBR image
mknod /tmp/hda
dd if=/mnt/source/pub/mbr.img of=/tmp/hda

%post
# Download a customized xorg.conf file via ftp -- the echo allows a
# DHCP client system to use hostnames in the %post section
echo nameserver 192.168.0.254 >> /etc/resolv.conf

lynx -source ftp://server1/pub/xorg.conf > /etc/X11/xorg.conf

# Suppress the rewriting of /etc/resolv.conf on a DHCP client
cat >> /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
PEERDNS=no
EOF
```

## End of Unit 7

- Questions and Answers
- Summary
  - Steps of the installation
  - Important Anaconda switches
  - **system-config-kickstart**
  - **ksvalidator**

7-20

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Lab 7

# Installation and System-Initialization

---

**Goal:** Successfully install Red Hat Enterprise Linux.

**System Setup:** A computer capable of booting from CD.

*Warning:* If your system cannot boot from the provided boot .iso CD, you might need to change the boot order in the computer's BIOS; however, please do not change BIOS settings without being instructed to do so.

## Sequence 1: Installing Red Hat Enterprise Linux

**Scenario:** A new server was just delivered. Before moving it into production, you plan to perform a test installation.

**Deliverable:** A basic system with a Volume Group which is mirrored.

**Instructions:**

1. As root, clear your partition table by executing the following:

```
# dd if=/dev/zero of=/dev/{s,h}da count=1000
```

2. Perform an NFS-based install, using the following installation information:

```
Install:      NFS
Network:      use DHCP, unselect IPv6
Server:       192.168.0.254
Mount point:  /var/ftp/pub
```

Use DHCP for your own network configuration. Use only IPv4 networking. Configure keyboard mapping and language as you prefer. Skip the installation number.

3. Partition the disk according to the following table:

```
/boot      128 MB
/           3096 MB
swap       512 MB
2 * RAID   2048 MB
```

Remember that RAID is a not a mount point; it is a filesystem type.

*Note:* If your hard disk is empty, you may receive a warning box. If asked to initialize the drive, Click Yes.

4. Combine the two RAID partitions into a RAID 1 device with “physical volume” as the filesystem type.

5. Create a volume group name “MyGroup” with the following logical volumes:

Name	Size	Mount Point
lv.home	128 MB	/home
lv.srv	512 MB	/srv
lv.tmp	256 MB	/tmp

6. Use the default Bootloader, Network, and Security Settings. Choose “redhat” as the root password.

7. Deselect GNOME Desktop (this will allow this temporary install to complete significantly faster). Install **system-config-kickstart** in addition to the default set of packages. This package is in **Base System/Administration Tools**. Do not add support for extra tasks. Add language

packages if required. After reboot complete the initial setup, but do not register with Red Hat Network. Check the log files for errors.

## Sequence 2: Kickstart Installation

**Deliverable:** A system that is automatically installed with kickstart.

**System Setup:** **system-config-kickstart** reads the list of available packages from RHN or yum repositories. Since the system is not currently connected to Red Hat Network, we must configure to use a yum repository on server1.

Put the following lines in `/etc/yum.repos.d/base.repo`:

```
[base]
name=Red Hat Enterprise Linux
baseurl=http://server1.example.com/pub/Server
enabled=1
gpgcheck=0
```

### Instructions:

1. Open `/root/anaconda-ks.cfg` in **system-config-kickstart**. Make sure that `server1:/var/ftp/pub` is used as **Installation Method**.
2. Partition the disk according to the following table:  
  

<code>/boot</code>	128 MB
<code>/</code>	4096 MB
<code>swap</code>	512 MB
<code>/home</code>	1024 MB
3. Select **Firewall Configuration**, and disable the default firewall. Leave SELinux active. Select **Display Configuration**, and check your color depth and resolution. Select **Package Selection**, highlight **Desktop Environments**, and check GNOME Desktop. Save the file and exit **system-config-kickstart**.
4. Use a text editor to add the line `"key --skip"` to the top of the kickstart file, just below the URL line, to avoid being prompted for the activation key. Define a student account with limited sudo access in your `%post` section. Below the partitioning information, add the following lines:  
  

```
%post
useradd student
echo password | passwd --stdin student
echo "student ALL=/bin/mount, /bin/umount" >> /etc/sudoers
```
5. Save the file, and check the contents by running it through **ksvalidator** to make sure that no syntax errors are in the file.

6. Start a web browser and visit the following url:

`http://server1.example.com/ksupload.html`

Follow the instructions there to upload your kickstart file to server1. When this is done, you will be given a **linux ks=...** command to use in the next step. Make a note of this command.

7. Reboot your system using boot media provided by the instructor. When the `boot :` prompt appears, use the **linux ks=...** instruction from the previous step.

*Note:* append **noipv6** to the command to deactivate IPv6 during installation. This will speed up the process of assigning your system an IP address significantly.

If anything is missing from the kickstart file, the installer will raise a dialog allowing you to add in the required information. You will use this installation for the remainder of this course.

## Sequence 1 Solutions

1. As root, clear your partition table by executing the following:

```
# dd if=/dev/zero of=/dev/{s,h}da count=1000
```

2. Perform an NFS-based install.

- a. Boot up system using CD
- b. Press *Enter* at the boot: prompt.
- c. Choose the appropriate language.
- d. Press *Enter* on the OK prompt.
- e. Choose the appropriate keyboard.
- f. Press *Enter* on the OK prompt.
- g. Choose **NFS image** for the installation method
- h. Configure TCP/IP. Select **Use dynamic IP configuration (BOOTP/DHCP)**. Deselect **Enable IPv6 support**.
- i. Press *Enter* on the OK prompt.
- j. Enter the appropriate information for an NFS installation  
  
NFS Server Name: **192.168.0.254**  
Red Hat Enterprise Linux directory: **/var/ftp/pub**
- k. At this point Anaconda (the installer) will retrieve the necessary installation image and will probe the system for its monitor and mouse type and will finally present you with the welcome screen. Click **Next**.
- l. Anaconda then checks if Red Hat Enterprise Linux is already installed on this system. If yes, it will offer an Upgrade instead of a fresh installation. Choose **Install Red Hat Enterprise Linux** Click **Next**.
- m. Select **Skip entering Installation Number**. Click **Next**. Confirm by clicking **Skip**.

3. Partition the disk according to the following table:

/boot	128 MB
/	3096 MB
swap	512 MB
2 * RAID	2048 MB

Remember that software RAID is a not a mount point; it is a filesystem type.



*Note:* If your hard disk is empty, you may receive a warning box. If asked to initialize the drive, Click Yes.

- a. Select **Create custom layout**. Click **Next**.
- b. To remove existing partitions, select the partition to delete and the click **Delete Partition**.

*Hint:* If you mark an entire disk you can delete multiple partitions at once. You must remove existing LVM volumes or RAID meta devices before deleting partitions.

- c. Click **New**.
  - d. Enter `/boot` as Mount Point. Use a fixed size of 128 MB. The filesystem type should be ext3 (default). Click **OK**.
  - e. Repeat the same process for a new / partition of 3096 MB.
  - f. Create the two RAID partitions. Leave the Mount Point empty and select RAID as the filesystem type and again a fixed size of 2048 MB.
  - g. Create a swap partition. Leave the Mount Point empty and select swap as the filesystem type. Set a fixed size of 512 MB.
4. Combine the two RAID partitions into a RAID 1 device with “physical volume (LVM)” as the filesystem type.
- a. Click **RAID**
  - b. Select Create a RAID device
  - c. Set the RAID level to RAID1.
  - d. Select the filesystem type physical volume (LVM)

5. Create a volume group named “MyGroup” with the following logical volumes:

Name	Size	Mount Point
lv.home	128 MB	/home
lv.srv	512 MB	/srv
lv.tmp	256 MB	/tmp

- a. Click **LVM**
- b. Set the Volume Group name to “MyGroup”
- c. Click **Add** to add a new logical volume
- d. Select the **Mount Point** /
- e. Enter “lv.home” as the **Logical Volume Name**

- f. Set the **Size (MB)** to 128
  - g. Click **OK** and repeat for the other volumes
6. Use the default Bootloader, Network, and Security Settings. Choose the appropriate locality settings and "redhat" as the root password.
  - a. Use the default Bootloader settings unless the instructor advises otherwise; do not create a Bootloader password.
  - b. Choose DHCP for networking and activate on boot
  - c. Set the time zone as appropriate for your location; implement UTC if the instructor suggests it
  - d. Set the root password to redhat (it is not a good password, but please use it anyway).
7. Deselect GNOME Desktop (this will allow this temporary install to complete significantly faster). Install **system-config-kickstart** in addition to the default set of packages. This package is in **Base System/Administration Tools**. Do not add support for extra tasks. Add language packages if required. After reboot complete the initial setup, but do not register with Red Hat Network. Check the log files for errors.
  - a. Select **Customize now** and click **Next**.
  - b. Select the **Desktop** tab. Deselect **GNOME Desktop**.
  - c. Select the **Base System** tab. Highlight **Administration Tools** and click **Optional packages**. Select **system-config-kickstart**.
  - d. Add language packages if required and click **Next**.
  - e. You should now be at the About to Install screen. Click **Next** to begin.
  - f. After the reboot following the installation, complete the initial set up tool. Create a user account of your choice. Do not register the machine with Red Hat Network. Select **No, I prefer to register at a later time** followed by **No thanks, I'll connect later**.
  - g. Choose enable Firewall, Leave SELinux at the default state Active.
  - h. Once you have completed the installation and the newly-installed system has booted, log in as root and examine the following:
    - /var/log/messages
    - /var/log/dmesg

## Sequence 2 Solutions

1. Open `/root/anaconda-ks.cfg` in **system-config-kickstart**. Make sure that `server1:/var/ftp/pub` is used as **Installation Method**.
  - a. Start **system-config-kickstart**.
  - b. Go to **File/Open File** and open the file `/root/anaconda-ks.cfg`
  - c. Select **Installation Method** and choose NFS as method. Set the **NFS Server** to `server1.example.com` and the **NFS Directory** to `/var/ftp/pub`.
2. Partition the disk according to the following table:

<code>/boot</code>	128 MB
<code>/</code>	4096 MB
<code>swap</code>	512 MB
<code>/home</code>	1024 MB

  - a. Go to **Partition Information**.
  - b. Select **Remove existing Linux partitions**.
  - c. Click on **Add**
  - d. Select `/boot` as **Mount Point**. Choose 128 MB as the **Size**.
  - e. Make sure that **Format partition** is selected.
  - f. Repeat with appropriate changes for the `/` file system and for the swap partition.
3. Select **Firewall Configuration**, and disable the default firewall. Leave SELinux active. Select **Display Configuration**, and check your color depth and resolution. Select **Package Selection**, highlight **Desktop Environments**, and check GNOME Desktop. Save the file and exit **system-config-kickstart**.
4. To avoid being prompted for an installation number during the kickstart, you may open `/root/ks.cfg` in a text editor and manually add the following line to the top of the file:

```
key --skip
```

This will limit the set of packages available to kickstart to the base repository for Server, however.

Define a student account with limited sudo access in your `%post` section. Below the partitioning information, add the following lines:

```
%post
useradd student
echo password | passwd --stdin student
echo "student ALL=/bin/mount, /bin/umount" >> /etc/sudoers
```

5. Save the file, and check the contents by running it through **ksvalidator** to make sure that no syntax errors are in the file.

Run the command:

```
# ksvalidator /root/ks.cfg.
```

If there is no output produced by the command, the kickstart file has successfully validated. This means the syntax of the file is correct, but kickstart may still fail due to logical problems in the file.

6. Start a web browser and visit the following url:

```
http://server1.example.com/ksupload.html
```

Follow the instructions there to upload your kickstart file to server1. When this is done, you will be given a **linux ks=...** command to use in the next step. Make a note of this command.

7. Reboot your system using boot media provided by the instructor. When the **boot :** prompt appears, use the **linux ks=...** instruction from the previous step.

*Note:* append **noipv6** to the command to deactivate IPv6 during installation. This will speed up the process of assigning your system an IP address significantly.

If anything is missing from the kickstart file, the installer will raise a dialog allowing you to add in the required information. You will use this installation for the remainder of this course.

## Unit 8

# System Initialization

8-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Discuss the boot sequence
- Understand GRUB's role
- Understand init's role
- Control System V services

8-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Boot Sequence Overview

- BIOS Initialization
- Bootloader
- Kernel initialization
- **init** starts and enters desired run level by executing:
  - `/etc/rc.d/rc.sysinit`
  - `/etc/rc.d/rc` and `/etc/rc.d/rc?.d/`
  - `/etc/rc.d/rc.local`
  - X Display Manager if appropriate

8-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Each major step in a Linux system's boot sequence - BIOS initialization, bootloader, kernel initialization, and init startup - is covered in the upcoming pages.

# BIOS Initialization

- Peripherals detected
- Boot device selected
- First sector of boot device read and executed

8-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## BIOS - Starting the boot process

The BIOS (Basic Input/Output System) is the interface between the hardware and software on a very basic level. The BIOS provides the basic set of instructions used by the operating system. A successful boot depends on the BIOS, which in fact provides the lowest level of interface to peripheral devices and controls.

The BIOS will first run a power on self test (POST), then it will look for peripherals and a device to boot from. The hardware configuration information is permanently stored in a small area (usually 64 bytes) of CMOS (Complementary Metal Oxide Semiconductor), most commonly referred to as simply "the CMOS". The CMOS is powered by a small battery located in your motherboard. This battery allows the CMOS to retain its settings even when the computer is turned off and disconnected from power.

At the end of the POST, a boot device is selected from the list of detected boot devices. Any modern BIOS will allow you to set the desired order of preference for the boot device from a list. Boot devices could include: the floppy drive, hard drive, CDROM, network-interface, Zip drive or other removable media).

The BIOS reads and executes the first physical sector of the chosen boot media on the system. Usually this is contained in the first 512 bytes of the hard disk.



# Bootloader Components

- **Bootloader**
  - 1st Stage - small, resides in MBR or boot sector
  - 2nd Stage - loaded from boot partition
- **Minimum specifications for Linux:**
  - Title, kernel location, OS root filesystem and location of the initial ramdisk (*initrd*)
- **Minimum specification for other OS:**
  - Title, boot device

8-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The bootloader is responsible for loading and starting your Linux operating system (or possibly other operating systems) when the computer is started up.

The bootloader is generally invoked in one of two ways:

- BIOS passes control to an initial program loader (IPL) installed within a drive's Master Boot Record.
- BIOS passes control to another bootloader, which passes control to an IPL installed within a partition's boot sector.

In either case, the IPL (initial program loader) must exist within a very small space, no larger than 446 bytes. Therefore, the IPL for GRUB is merely a first stage, whose sole task is to locate and load a second stage bootloader, which does most of the work to boot the system.

There are two possible ways to configure bootloaders:

- **primary bootloader:** Install the first stage of your Linux bootloader into the Master Boot Record. The bootloader must be configured to pass control to any other desired operating systems.
- **secondary bootloader:** Install the first stage of your Linux bootloader into the boot sector of some partition. Another bootloader must be installed into the MBR, and configured to pass control to your Linux bootloader.

# GRUB and grub.conf

- GRUB “the GRand Unified Bootloader”
  - Command-line interface available at boot prompt
  - Boot from ext2/ext3, ReiserFS, JFS, FAT, minix, or FFS file systems
  - Supports MD5 password protection
- /boot/grub/grub.conf
- Changes to grub.conf take effect immediately
- If MBR on /dev/hda is corrupted, reinstall the first stage bootloader with:
  - /sbin/grub-install /dev/hda

8-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

/boot/grub/grub.conf has a format of global options followed by boot stanzas. Here is a sample grub.conf:

```
timeout=5
splashimage=(hd0,0)/grub/splash.xpm
hiddenmenu
password --md5 $1$/iX9y$Bk4yt37Ch2fZ5GFA
default=0
```

```
title Red Hat Enterprise Linux AS (2.6.9-648_EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-648.EL ro root=/dev/VolGroup00/LogVol100 rhgb quiet
    initrd /initrd-2.6.9-648.EL.img
```

```
title Windows XP Pro
    rootnoverify (hd0,1)
    chainloader +1
```

Changes to grub.conf take effect immediately. GRUB reads the configuration file at boot time, so the grub.conf file must be stored on a filesystem GRUB understands. These include ext2/ext3, reiserfs, FAT, minix, and FFS.

The MD5 password hash can be created with **grub-md5-crypt**.

If for some reason your MBR becomes corrupted and you need to reinstall GRUB, you can do so with the command **/sbin/grub-install boot-device**. Occasionally it may prove necessary for the user to set up grub manually. If **grub-install** fails for some reason try the following:

1. Type the command **grub** and press *Enter*

2. Type **root (hd0,0)**

3. Type **setup (hd0)**

4. Type **quit**

# Starting the Boot Process: GRUB

- Image selection
  - Select with space followed by up/down arrows on the boot splash screen
- Argument passing
  - Change an existing stanza in menu editing mode
  - Issue boot commands interactively on the GRUB command line

8-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## The GRUB Boot Screen

When GRUB starts up, a graphical splash screen can be accessed by pressing *Return*; or *Space*. 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.

## Menu Editing Mode

If you then select an entry and type *e*, you will be dropped into menu editing mode. This mode allows you to modify an existing boot stanza to pass options to the kernel or *init*, or select alternate root file systems or kernel files than you have configured in your existing stanzas. You can use arrow keys to select a line, *e* to edit a line, *d* to delete a line, *o* to add a line, and *b* to boot. For example, to boot into runlevel 2, you could select menu editing mode, select your Linux boot stanza, add a 2 to the end of your kernel line, and type *b* to boot the modified menu entry.

## The GRUB Command Line

GRUB provides a command-line interface which can be used to write a temporary boot command from scratch, view the contents of files on the filesystem, perform diagnostic tests, or experiment with GRUB configurations. Most commands supported by the configuration file are available for interactive use. Editing commands are similar to those used by the bash shell, and *Tab* completion is available. If GRUB is not able to find a valid *grub.conf* file, it will default to the command line.

To exit menu editing mode or the command line and go back to the main GRUB menu, type *Esc*.

For more information about GRUB and *grub.conf*, look at **info grub**.

# The Chicken/Egg Module Problem and the Initial RAM Disk

- To mount the root filesystem, the kernel typically needs to load modules
  - Examples: ext3, jbd, raid1, scsi\_mod
- An *initial RAM disk* provides modules
  - Compressed cpio archive containing modules, other material
  - Created at install time
  - Specific to a particular hardware and software platform
  - Made available to the kernel by GRUB
- Use **mkinitrd** to rebuild
  - Example:

```
mkinitrd /boot/initrd-$(uname -r).img $(uname -r)
```

8-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The modular nature of the Linux kernel presents a boot time issue:

The Linux kernel needs to mount the root filesystem. To do so, it typically needs access to modules. For example, the following filesystem-related capabilities are modular (not built into the basic kernel):

- ext3 filesystems and journaling for ext3 filesystems
- Logical volume management
- Software RAID
- SCSI controller support

If the kernel requires any of these capabilities to mount the root filesystem, it will need access to the related modules.

But this presents a quandary: where does the kernel get the modules? Typically, the kernel gets the modules from the `/lib/modules/$(uname -r)` directory. But, as the root filesystem is not yet mounted, it cannot access modules in this location. Hence, this presents a classic chicken/egg problem: the kernel cannot mount the root filesystem without access to the modules and the modules are on the root filesystem.

The GRUB bootloader and the kernel work together to provide the solution to this problem through the use of the initial RAM disk, which is part of the typical GRUB specification for a Linux kernel:

```
title Red Hat Enterprise Linux ES (2.6.9-22.0.1.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-22.0.1.EL ro root=LABEL=/
    initrd /initrd-2.6.9-22.0.1.EL.img
```

The last line indicates that the initial RAM disk file is called `initrd-2.6.9-22.0.1.EL.img` (and that it is resident in `/boot`). Under Red Hat Enterprise Linux, version 4, this file contains a compressed `cpio` archive containing, among other things, modules needed to mount filesystems.

An initial RAM disk is specific to a particular hardware and software platform. Typically, it is created at install time and only those modules needed to mount filesystems for that system are included.

Initial RAM disks need to be recreated when filesystem hardware or software changes. For example, if a system has `ext2` filesystems that have been converted to `ext3` filesystems, the initial RAM disk will need to be recreated. To create an initial RAM disk, use the **`mkinitrd`** command:

```
mkinitrd /boot/initrd-$(uname -r).img $(uname -r)
```

The first argument is the path name to the initial RAM disk; the second argument is the directory within the `/lib/modules` directory that contains the modules. As initial RAM disks are specific to particular kernels, it is essential that the modules are appropriate for that kernel.

It is also possible to force a particular module to be placed in an initial RAM disk. This may be useful if, for example, you intend to make changes that you have not yet implemented. Two methods can be used to force a module into the initial RAM disk:

1. Using the **`--with`** option:

```
mkinitrd --with=scsi_mod /boot/initrd-$(uname -r).img $(uname -r)
```

Neither the `.ko` suffix nor the full path name to the module are necessary.

2. Placing a filesystem-related directive in `/etc/modprobe.conf`:

```
echo "alias scsi_hostadapter qla2300" >> /etc/modprobe.conf
mkinitrd /boot/initrd-$(uname -r).img $(uname -r)
```

The commands above will result in at least the `qla2300` module (plus any dependent modules) to be loaded into the initial RAM disk.

# Kernel Initialization

- Kernel boot time functions
  - Device detection
  - Device driver initialization
  - Mounts root filesystem read only
  - Loads initial process (init)

8-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *Examining kernel initialization*

The kernel initialization files generate good output, but scroll by quickly. A good way to examine this output is to view `/var/log/dmesg`, which contains a snapshot of these kernel messages taken just after control is passed to `init`. Review of this output will reveal the basic initialization steps of the Linux kernel:

Device drivers compiled into the kernel are called, and will attempt to locate their corresponding devices. If successful in locating the device, the driver will initialize and usually log output to the kernel message buffer.

If essential (needed for boot) drivers have been compiled as modules instead of into the kernel, then they must be included in an `initrd` image, which is then temporarily mounted by the kernel on a RAM disk to make the modules available for the initialization process.

After all the essential drivers are loaded, the kernel will mount the root filesystem read-only.

The first process is then loaded (`init`) and control is passed from the kernel to that process.

# init Initialization

- init reads its config: `/etc/inittab`
  - initial run level
  - system initialization scripts
  - run level specific script directories
  - trap certain key sequences
  - define UPS power fail / restore scripts
  - spawn gettys on virtual consoles
  - initialize X in run level 5

8-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *init*

init is the parent of all processes. This is easily shown by running the **ps**tree command:

```
[student@stationX]$ pstree
init--+-apmd
      |
      |--atd
      |--automount
      |--crond---crond
      |--deskguide_apple
      |--gdm--+-X
              |--gdm---gnome-session
```

Because init is the first process, it will always have a PID of number 1.

The file `/etc/inittab` contains the information on how init should set up the system in every run level, as well as the run level to use as default.

If the `/etc/inittab` file is missing or seriously corrupt, you will not be able to boot to to any of the standard run levels (0-6) and will need to use single or emergency mode instead. This procedure is discussed in depth in the Troubleshooting unit of this course.



# Run Levels

- init defines run levels 0-6, S, emergency
- The run level is selected by either
  - the default in `/etc/inittab` at boot
  - passing an argument from the bootloader
  - using the command `init new_runlevel`
- Show current and previous run levels
  - `/sbin/runlevel`

8-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The following chart details the run levels that Linux defines by default:

0	halt
1	single user mode
2	multiuser mode without NFS
3	full multiuser mode, typical for servers
4	officially undefined
5	graphical login, typical for desktops and laptops
6	reboot
s, S, single	alternate single user mode
emergency	bypass <b>rc.sysinit</b> , <b>su</b> login

## `/etc/rc.d/rc.sysinit`

- Important tasks include:
  - Activate udev and `selinux`
  - Sets kernel parameters in `/etc/sysctl.conf`
  - Sets the system clock
  - Loads keymaps
  - Enables swap partitions
  - Sets hostname
  - Root filesystem check and remount
  - Activate RAID and LVM devices
  - Enable disk quotas
  - Check and mount other filesystems
  - Cleans up stale locks and PID files

8-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

When `rc.sysinit` first starts, it prompts you to press the letter `i` if you want to enter interactive mode. In interactive mode, it will prompt you for confirmation before performing most of its functions. The script is straightforward in layout, and well documented, so for additional details simply look at the script source directly.

## /etc/rc.d/rc

- Initializes the default run level per the /etc/inittab file initdefault line such as id:3:initdefault:
- 10:0:wait:/etc/rc.d/rc 0
- 11:1:wait:/etc/rc.d/rc 1
- 12:2:wait:/etc/rc.d/rc 2
- 13:3:wait:/etc/rc.d/rc 3 (default)
- 14:4:wait:/etc/rc.d/rc 4
- 15:5:wait:/etc/rc.d/rc 5
- 16:6:wait:/etc/rc.d/rc 6

8-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The rc script initializes the intended run level as defined by the initdefault line in the /etc/inittab and it is responsible for starting/stopping services when the runlevel changes.

# System V run levels

- Run level defines which services to start
  - Each run level has a corresponding directory:
    - `/etc/rc.d/rcX.d`
  - The System V init scripts reside in:
    - `/etc/rc.d/init.d`
  - Symbolic links in the run level directories call the `init.d` scripts with a start or stop argument

8-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The ability to change run levels allows easy interaction for administrators. If you need to test some software or perform system maintenance you can switch to run level 1 and the system will be in single-user mode. You do not have to shutdown and restart to change run levels. The scripts that launch the system services are located in the `/etc/rc.d/init.d`.

```
$ ls -l /etc/rc.d/init.d
```

```
-rwxr-xr-x 1 root root 1535 Jul 15 12:09 amd
-rwxr-xr-x 1 root root  798 Aug  4 03:01 anacron
-rwxr-xr-x 1 root root 1289 Aug 17 03:15 apmd
-rwxr-xr-x 1 root root  908 Aug 11 15:43 arpwatc
...      output truncated      ...
```

```
$ ls -l /etc/rc.d/rc3.d
```

```
lrwxrwxrwx 1 root root  14 Sep 22 16:15 K15pvmd -> ../init.d/pvmd
lrwxrwxrwx 1 root root  13 Sep 22 16:11 K20nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root  15 Sep 22 16:09 S05kudzu -> ../init.d/kudzu
lrwxrwxrwx 1 root root  18 Sep 22 16:08 S08ipchains -> ../init.d/ipchains
lrwxrwxrwx 1 root root  17 Sep 22 16:04 S10network -> ../init.d/network
lrwxrwxrwx 1 root root  16 Sep 22 16:04 S12syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root  17 Sep 22 16:11 S13portmap -> ../init.d/portmap
...      output truncated      ...
```

## `/etc/rc.d/rc.local`

- Run after the run level specific scripts
- Common place for custom modification
- In most cases it is recommended that you create a System V *init* script in
- `/etc/rc.d/init.d` unless the service you are starting is so trivial it doesn't warrant it. Existing scripts can be used as a starting point.

8-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

### **rc.local - Final System V Initialization**

Because the **rc.local** script is run each time the system enters a run level, it is a convenient place to start processes that need to be running.

# Controlling Services

- Utilities to control default service startup
  - **system-config-services**: graphical utility that requires an X interface
  - **ntsysv**: ncurses based utility usable in virtual consoles
  - **chkconfig**: a fast, versatile command line utility that works well and is usable with scripts and Kickstart installations
- Utilities to control services manually
  - **service**: immediately start or stop a standalone service
  - **chkconfig** immediately starts and stops xinetd-managed services

8-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The job of the System V initialization scripts is to start services at boot time. Most of these services run as daemons, such as cups, cron and sendmail. Red Hat Enterprise Linux includes several utilities that facilitate the management of System V initialization.

**system-config-services** is an X client that presents a display of each of the services that are started and stopped at each run level. Services can be added, deleted, or re-ordered in run levels 3 through 5 with this utility.

**ntsysv** is a console-based interactive utility that allows you to control what services run when entering a given run level. This utility is used during system installation, but can be run from the command line. It configures the current run level by default. By using the `--level` option you can configure other run levels.

**chkconfig** is a command-line utility. When passed the `--list` switch, it displays a list of all System V scripts and whether each one is turned on or off at each run level. Scripts can be managed at each run level with the `on` and `off` `chkconfig` directives. The `--level` option can be used to specify the runlevels affected if the defaults are unacceptable.

The **service** command is used to start or stop a standalone service immediately; most services accept the arguments `start`, `stop`, `restart`, `reload`, `condrestart`, and `status` as a minimum.

The **system-config-services** and **chkconfig** commands will start or stop an xinetd-managed service as soon as you configure it on or off. Standalone services will not start or stop until the system is rebooted or you use the `service` command.

## End of Unit 8

- Questions and Answers
- Summary
  - System BIOS
  - GRUB
  - init
  - chkconfig and service

8-17

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 526 2994 or +1 (919) 754 3700.

# Lab 8

## Managing Startup

---

Goal: To familiarize yourself with the startup process

System Setup: A system installed with Red Hat Enterprise Linux



## Sequence 1: Changing the default run level

Deliverable: A system that boots to runlevel 3 by default.

### Instructions:

1. Change the default runlevel to level 3 and reboot.
2. Change the default run level back to level 5 and reboot.

## Sequence 2: Exploring an initial RAM disk

**Scenario:** An initial RAM disk is an essential element in the boot up process. In this sequence, we will investigate the contents of an initial RAM disk, modify the contents, and then boot using the new RAM disk.

**Deliverable:** A system booting with a new RAM disk.

### Instructions:

1. First, we begin by identifying your current initial RAM disk file. This file will be in `/boot` and it will contain the release level of the kernel in the name. That is, the file will have the form: `/boot/initrd-$(uname -r).img`. Display a long listing of this file on your system.

2. Run the **file** command against this file to further identify it. You will see that it is a file compressed using **gzip**.

Uncompress the initial RAM disk file and run the **file** command against this data.

*Warning:* Do not modify the initial RAM disk file itself!

You will see that the compressed file contains a **cpio** archive.

*Hint:* To accomplish this, consider that the **zcat** command will uncompress a stream of standard input, sending the uncompressed data to the standard output. Also consider that the **file** command will read from standard input if given a dash (-) as an argument.

3. List the contents of the compressed **cpio** archive. Note that within the `lib` subdirectory are a series of kernel objects (files ending in `.ko`). This should include `ext3.ko` (the module for `ext3` filesystems), `jbd.ko` (the module for journaling for `ext3` filesystems), and, depending on your hardware, may include other modules as well.

List the modules currently loaded into your running kernel. Modules are listed from most recently inserted to least recently inserted. Note that the modules on the bottom of the list are the modules that were loaded from your initial RAM disk.

4. Now, create a new initial RAM disk file, forcing the `raid1.ko` module to be included. Name the new initial RAM disk `/boot/initrd-raid1-$(uname -r).img`. Note that the `raid1.ko` module is not one of the modules in the original initial RAM disk file.

*Warning:* Be sure to give this file a new name; do not overwrite your current initial RAM disk!

5. In the `/boot/grub/grub.conf` file, copy the stanza for your kernel (the title, root, kernel, and `initrd` lines) so that you now have two versions of this entry. On the copy (without modifying the original stanza) modify the `initrd` line to use your new `initrd-raid1-$(uname -r).img` file. Note that you must actually list the version number; you cannot use the `$(uname -r)` syntax.

Reboot your computer, using the new initial RAM disk. Once again, list the modules in your currently running kernel. You should see the `raid1.ko` module now listed as one of the modules now available.

*Mandatory cleanup:* Reboot your computer such that you are using the original initial RAM disk.

## Sequence 3: GRUB

### Instructions:

1. Use GRUB at boot time to bring up Linux in various run levels. Boot the system into single user mode. Try out other runlevels as well.

## Sequence 1 Solutions

1. Change the default runlevel to level 3 and reboot.
  - a. Edit the `/etc/inittab` file and change the default run level as shown below from level 5 to level 3.

```
id:3:initdefault:
```

- b. Reboot the system.
2. Change the default run level back to level 5 and reboot.

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

## Sequence 2 Solutions

1. First, we begin by identifying your current initial RAM disk file. This file will be in `/boot` and it will contain the release level of the kernel in the name. That is, the file will have the form: `/boot/initrd-$(uname -r).img`. Display a long listing of this file on your system.

```
$ ls -l /boot/initrd-$(uname -r).img
```

2. Run the `file` command against this file to further identify it. You will see that it is a file compressed using `gzip`.

```
$ cd /boot
$ file initrd-$(uname -r).img
```

Uncompress the initial RAM disk file and run the `file` command against this data. *Warning:* Do not modify the initial RAM disk file itself!

```
$ zcat initrd-$(uname -r).img | file -
```

You will see that the compressed file contains a `cpio` archive.

3. List the contents of the compressed `cpio` archive. Note that within the `lib` subdirectory are a series of kernel objects (files ending in `.ko`). This should include `ext3.ko` (the module for `ext3` filesystems), `jbd.ko` (the module for journaling for `ext3` filesystems), and, depending on your hardware, may include other modules as well.

```
$ zcat initrd-$(uname -r).img | cpio -itv | less
```

List the modules currently loaded into your running kernel. Modules are listed from most recently inserted to least recently inserted. Note that the modules on the bottom of the list are the modules that were loaded from your initial RAM disk.

```
$ lsmod | less
```

4. Now, create a new initial RAM disk file, forcing the `raid1.ko` module to be included. Name the new initial RAM disk `/boot/initrd-raid1-$(uname -r).img`. Note that the `raid1.ko` module is not one of the modules in the original initial RAM disk file.

*Warning:* Be sure to give this file a new name; do not overwrite your current initial RAM disk!

```
$ mkinitrd --with=raid1 initrd-raid1-$(uname -r).img $(uname -r)
```

5. In the `/boot/grub/grub.conf` file, copy the stanza for your kernel (the title, root, kernel, and `initrd` lines) so that you now have two versions of this entry. On the

copy (without modifying the original stanza) modify the `initrd` line to use your new `initrd-raid1-$(uname -r).img` file. Note that you must actually list the version number; you cannot use the `$(uname -r)` syntax.

Your new stanza should look something like this:

```
title Red Hat Enterprise Linux ES (2.6.9-22.0.1.EL)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-22.0.1.EL ro root=LABEL=/
    initrd /initrd-raid1-2.6.9-22.0.1.EL.img
```

Reboot your computer, using the new initial RAM disk. Once again, list the modules in your currently running kernel. You should see the `raid1.ko` module now listed as one of the modules now available.

```
$ lsmod | less
```

*Mandatory cleanup:* Reboot your computer such that you are using the original initial RAM disk.

## Sequence 3 Solutions

1. Use GRUB at boot time to bring up Linux in various run levels. Boot the system into single user mode. Try out other runlevels as well.
  - a. Reboot Linux so that GRUB appears on your screen. If you have specified a "timeout=" value in `grub.conf` you will notice that the timer is counting down.
  - b. Before the timer counts down to zero, press the *space* to halt the timer.
  - c. Take note of the help text in the lower part of the GRUB display. Use the up/down arrow keys to navigate to the kernel you wish to boot. Then press *e* to override the contents of `grub.conf` for this kernel.
  - d. Once again, take note of the help text in the lower portion of the GRUB display. Use the up/down arrows to navigate to the line starting with the text **kernel** and press the *e* key.
  - e. You are now in GRUB edit mode with the cursor at the end of the line. Press the space bar followed by the *s* key, then press the *Enter* key. You will note that the GRUB display returns to the prior screen and now has the new text "S" appended to the kernel line. If you wish to undo all changes you have made in GRUB, simply press the *Esc* key to return to the prior screen.
  - f. Press the *b* key to boot using these GRUB options. In this example, you will come up in runlevel "S" or single user.
  - g. Review the contents of the `grub.conf` file. You will note that the change you made at the GRUB screens did not update this file.
  - h. Repeat the steps above, trying different runlevels such as "emergency", "1", etc.



## Unit 9

# RPM, YUM, RHN

9-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Objectives

Upon completion of this unit, you should be able to:

- Understand RPM
- Examine significant options
- Query packages with **rpm -q**
- Understand yum
- Add a yum repository
- Configure and use Red Hat Network

9-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# RPM Package Manager

- RPM Components
  - local database
  - **rpm** and related executables
  - RPM front-ends such as yum
  - package files
- Primary Functions
  - install/remove
  - query
  - verify

9-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The RPM Package Manager greatly simplifies the distribution, installation, upgrading, and removal of software on Red Hat Enterprise Linux systems. The RPM system consists of a local database, the **rpm** executable, **rpm** package files.

The local RPM database is maintained in `/var/lib/rpm`. The database stores information about installed packages such as file attributes and package prerequisites. An administrator rarely, if ever, modifies the database directly, but instead uses the **rpm** command.

Software to be installed using **rpm** is distributed through **rpm** package files, which are essentially compressed archives of files and associated dependency information. Package files are named using the following format:

`name-version-release.architecture.rpm`

The **version** refers to the open source version of the project, while the **release** refers to Red Hat internal patches to the open source code.

In contrast to package management on some other platforms, RPM's design does not provide interactive configuration of software as part of the package load process. RPM can perform configuration actions as part of the installation, but these are scripted, not interactive. It is common for packages to install with reasonable default configurations applying. On the other hand, some software installs in an unconfigured state.

**rpm** is a back-end for other programs such as **yum** or **system-config-packages**. These tools provide significant advantages such as automatic dependency resolution.

# Installing and Removing Software

- Primary RPM options:
  - Install: **rpm -i, --install**
  - Upgrade: **rpm -U, --upgrade**
  - Freshen: **rpm -F, --freshen**
  - Erase: **rpm -e, --erase**
- Output Options: **-v, -h**
- URL support: **ftp://** (with globbing), **http://**
- Many other install-options are available to address special cases.

9-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Installing: rpm -i

The primary function of RPM is to install, upgrade, and remove software from a system. A package is installed using a command such as **rpm -i zip-2.3-8.i386.rpm**.

When installing an rpm package, the rpm command will consult the local database to ensure that (1) any prerequisites (in the form of files, libraries, rpms, or generally defined “provisions”) are installed on the system, and (2) installing the rpm will not clobber any preexisting files. The checks can be omitted by enabling the **--nodeps** or **--replacefiles** command-line switches, respectively, or both using the **--force** switch. rpm will provide “pretty” output if called with the **-v** (print package name) and **-h** (print hash marks) options.

## Upgrading: rpm -U and rpm -F

rpm can be used to upgrade already installed software with the **-U (--upgrade)** command-line switch. When upgrading, the original package (with the exception of configuration files) on the system will be removed, and the new package installed. Configuration files from the original installation are saved with a **.rpmsave** extension.

Freshening is almost identical to upgrading, except when the package specified on the command line is not already installed on the system. When upgrading with **-U**, the package will be installed whether or not it is already installed; when freshening, the package will be ignored if not already installed. To apply all errata released by Red Hat for all packages installed on your system, ignoring errata for uninstalled packages, execute the following:

## Uninstalling: rpm -e

Software is removed from your system using the **-e (--erase)** command-line switch. The package argument must be the installed package's name, not the package file name. For example, these commands first install the zip package file, and then remove it:

**rpm -ihv zip-2.3-8.i386.rpm**

**rpm -e zip**

# Updating a Kernel RPM

- Make sure to install kernel updates
- Do not use **rpm -U** or **rpm -F** !
  - **rpm -ivh kernel-version.arch.rpm**
  - Boot new kernel to test
  - Revert to old kernel if a problem arises
  - **rpm -e kernel-oldversion** if no problems

9-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Installing a new kernel is one of the few things you will do on your system that requires a reboot operating systems. It also requires a little more thought and caution, as it is quite simple to render a system temporarily inoperable if you are careless when updating the kernel. Unlike just about any other upgrade you might do, you should NOT upgrade the kernel using **rpm -U** or **-F**.

Recall how **rpm** functions with the **-U** and **-F** options: it determines whether a version already exists on the system, and if so, whether the version to be installed is newer. If it is newer, **rpm** first installs the new version, thereby replacing old files -- except those marked as configuration files. It then removes the old package, but only deletes files that do not exist in the new version.

Because upgrading removes the previous kernel version, if your newly-installed kernel proves unstable, you could be left with an unbootable system. You would have to resort to alternate boot media such as a boot floppy or the CD-ROM. When you run an install instead of an upgrade, the old version of the kernel is still available and can be selected from the bootloader.

In addition, kernel modules are version-specific, and an upgrade will remove all modules that your present kernel is using, leaving the system unable to dynamically load device drivers or other modules.

Because all of the kernel RPM's files are version-specific (that is, they either include version information in their names or are stored in version-specific paths), it is possible to install multiple versions of the kernel package. If you use **rpm -ivh**, instead of **-U**, then the new kernel will be added to your system, but your old kernel remains on it as well.

By default, the new kernel is automatically added to GRUB. You can change this behavior by editing `/etc/sysconfig/kernel`.

# rpm Queries

- Syntax:
  - **rpm -q what\_packages what\_information**
- Installed Package Options:
  - **rpm -qa** lists installed packages
  - **rpm -qf filename#**shows owning package
  - **rpm -qi package\_name#**general information
  - **rpm -ql package\_name#**lists files in package
- Uninstalled Package Options:
  - **rpm -qip package\_file.i386.rpm**
  - **rpm -qlp package\_file.i686.rpm**

9-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

RPM provides robust querying, which is invoked with **rpm -q** or **rpmquery**. Query options fall into one of two categories: those that specify which packages to query, and those that specify what information to retrieve. The first must be specified; the second defaults to the package name.

Here's a list of common package specification parameters and what they return:

<b>-qa</b>	all installed packages
<b>-q packagename</b>	the named package and version
<b>-qf filename</b>	the package that owns the file
<b>-qp package_file_name</b>	the (possibly uninstalled) package file
<b>--whatrequires capability</b>	all packages that require capability
<b>--whatprovides capability</b>	all packages that provide capability

The following is a list of common query information parameters and what they return:

<b>-i</b>	general package information
<b>-l</b>	package files
<b>--requires</b>	package prerequisites
<b>--provides</b>	capabilities provided by package
<b>--scripts</b>	scripts run upon installation and removal
<b>--changelog</b>	package revision history
<b>--queryformat</b>	format custom-formatted information (use <b>rpm --querytags</b> to list available tags for use in format string)

# **rpm Verification**

- Installed RPM File Verification:
  - **rpm -V <package\_name>**
  - **rpm -Vp <package\_file>.i386.rpm**
  - **rpm -Va**
- Signature verification BEFORE package install:
  - **rpm --import RPM-GPG-KEY**
  - **rpm -K <package\_file>.i386.rpm**

9-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## **RPM file verification**

Verifying an installed package compares the file sizes, permissions, type, owner, group, MD5 checksum, and modify time against the RPM database. Any inconsistencies will be reported. An installed package can also be verified against a package file as well:

**rpm -V zip** - verifies the installed zip rpm against the RPM database

**rpm -Va** - verifies all installed RPMS against the RPM database

**rpm -Vp zip-2.3-8.i386.rpm** - verifies the installed zip package against the zip package file

## **RPM signature verification**

Red Hat signs all package files with a GPG private signature. The complementary public signature is shipped with every Red Hat distribution. To verify the integrity of any package file, you must first import the Red Hat public key. The rpm utility will automatically verify the signature of any package you install at install time. You can also check the integrity of package files using the **--checksig** option.

**rpm --import /mnt/cdrom/RPM-GPG-KEY**

**rpm -qa gpg-pubkey**



# About yum

- Front-end to **rpm**
  - Designed to resolve package dependencies
  - Can locate packages across multiple repositories
- Replacement for **up2date**

9-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Development of RPM cemented the future of Linux by greatly simplifying installation of software. As the operating system became more complex, RPM began to show a few weaknesses, primarily its inability to resolve dependencies.

```
[root@stationX ~]# rpm -ivh x3270-x11-*
warning: x3270-x11-3.3.4p7-3.el5.1.x86_64.rpm: Header V3 DSA signature: NOKEY
key ID 897da07a
error: Failed dependencies:
        x3270 = 3.3.4p7 is needed by x3270-x11-3.3.4p7-3.el5.1.x86_64
[root@stationX Server]#
```

Unfortunately, there is no way to look at the above output and determine if the suggested RPM, x3270, also has a dependency. As a matter of fact it is possible that dozens of RPMs would have to be installed. If the need RPMs were not available in the current directory, it would be up to the user to located and install each.

To solve the problem of dependency resolution and package location, volunteer programmers at Duke University developed Yellow dog Update, Modified, or YUM for short. The system is based on repositories that hold RPMs and a repodata file list. The yum application can call upon several repositories for dependency resolution, fetch the RPMs, and install the needed packages.

```
[root@stationX ~]# yum install x3270-x11
... output truncated ...
Dependencies Resolved
```

```
=====
Package                        Arch      Version      Repository    Size
=====
Installing:
x3270-x11                      x86_64     3.3.4p7-3.el5.1  server1       424 k
Installing for dependencies:
x3270                          x86_64     3.3.4p7-3.el5.1  server1       142 k
```

Transaction Summary

```
=====
Install      2 Package(s)
```

Update 0 Package(s)  
Remove 0 Package(s)

Total download size: 566 k  
Is this ok [y/N]:

# Using yum

- Install/Remove/Update
  - **yum install** *package...*
  - **yum remove** *package...*
  - **yum update** [*package...*]

9-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The command-line utility **yum** gives you an easy way to manage the packages on your system:

```
[root@stationX ~]# yum install firefox
```

The above command will search the configured repositories for a package named **firefox**, and if found will install the latest version, pulling in dependencies if needed.

```
[root@stationX ~]# yum remove mypackage
```

The above command will try to remove the package named **mypackage** from your system. If any other package depends on **mypackage** **yum** will prompt you about this, giving you the option to remove those packages as well.

```
[root@stationX ~]# yum update [mypackage...]
```

If any packages are specified on the command-line **yum** will search the configured repositories for updated versions of those packages and install them. When no packages are specified **yum** will search for updates to all of your currently installed packages.

# Searching packages/files

- Searching packages
  - **yum search** *searchterm*
  - **yum list** (*all/available/extras/installed/recent/updates*)
  - **yum info** *packagename*
- Searching files
  - **yum whatprovides** *filename*

9-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**yum search** *searchterm* will search all known package names and descriptions for *searchterm*:

```
[root@stationX ~]# yum search cairo
```

**yum list** *searchterm* will search all known package names for *searchterm*. *searchterm* can include wildcards:

```
[root@stationX ~]# yum list '*irefo*'
```

**yum info** *package...* will search all the package database for *package* and display some info about the package.

```
[root@stationX ~]# yum info '*irefo*'
```

**yum whatprovides** *filename* will search all packages (both installed and available) for *filename*. This can be useful when you know the filename of an executable/library you need, but you don't know the package name.

```
[root@stationX ~]# yum whatprovides /usr/sbin/sendmail
```

## Configuring Additional Repositories

- Create a file in `/etc/yum.repos.d` for your repository
- Required information
  - `[repo-name]`
  - `name=A nice description`
  - `baseurl=http://yourserver.com/path/to/repo`
  - `enabled=1`
  - `gpgcheck=1`
- Repository information is cached. To clear the cache:
  - `yum clean dbcache/all`

9-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Inside a repository declaration you can use variables like `$releasever` and `$basearch` to be substituted with the relevant information for your installation. Using this you can roll out one repository-file for use on multiple architectures or releases.

# Red Hat Network

- Centralized platform for systems management
  - provides Red Hat software packages
  - shows if errata are available for systems
  - can update many systems at once
  - allows full life cycle management
- Web based management interface
- Uses HTTPS for all transactions

9-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Red Hat Network allows administrators to manage software installation and upgrades efficiently using a combination of your RHN account and the **up2date** utility.

In March of 2001, the Lion worm, a self-spreading program intended to compromise security on Linux systems, made headlines. According to a report found at <http://www.sans.org>, the worm makes use of a "BIND vulnerability... that was reported back on January 29th, 2001". Red Hat had posted an updated version of the bind RPM that same day in January. If all Linux administrators had promptly updated their systems, the Lion worm would not have made headlines. Red Hat Network attempts to simplify the task of keeping software updated, reducing a system's vulnerability to exploits.

# Red Hat Network Server

- `rhn.redhat.com` or local Satellite/Proxy
  - Web based management of machines
  - RHN Proxy caches RHN traffic
  - RHN Satellite provides an autonomous RHN
- RHN Accounts
  - RHN Users for registration of machines and web based management
  - System ID for automatic authentication of systems

9-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

RHN Service is provided either by `rhn.redhat.com` or a local Satellite or Proxy server. The server provides the packages and offers remote management of the clients. Transactions (i.e. installation of a package) can be scheduled on the web interface, which are executed by clients the next time they poll RHN.

## RHN Proxy Server

Software updates and errata may be locally cached using a RHN Proxy server. Client profiles are still maintained on RHN servers. The base channel (i.e., "Red Hat Enterprise Linux (v. 5 for 32-bit x86)") is managed by RHN, but additional private sub-channels that allow the local distribution of custom software can be defined and locally administered. All interactions with RHN are mediated by the RHN Proxy Server, so only the proxy server needs Internet access.

## RHN Satellite Server

The RHN Satellite Server allows all aspects to be managed locally, including client profiles and custom channel management. Systems can be provisioned from bare metal with the Provisioning Entitlement. System management is performed using a local web server, and a local database maintains client accounts and profiles. The Satellite Server allows complete channel definition, control, and management. No Internet access is required.

## Accounts

Users login to the RHN web interface to manage the systems online. The RHN User is also used for registration of new systems. Upon registration a system-id is generated for the system, so that the system can connect to RHN non-interactively.

# Entitlements

- Grant access to software channels
  - Base Channel
  - Child Channel(s)
- Define level of service
  - Update
  - Management
  - Provisioning
  - Monitoring

9-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Channels

Entitlements define which software channels can be subscribed by a system. There are two different types of software channels. Systems are assigned to one base channel like Red Hat Enterprise Linux (v.5 for 32bit x86).

Child channels may be subscribed in addition. The Fast Track Child Channel, which is available to all systems provides bug fix errata before they are integrated into a maintenance release.

## Level of Service

Update Entitlements are included with all Red Hat Enterprise Linux subscriptions. They allow management of machines by a single administrator

Management Entitlements allow for the grouping of client systems, including collective software management and errata notifications. Multiple administrators may be defined and assigned to the various system groups.

With Provisioning Entitlements configuration files and Kickstart profiles can be managed.

Systems with Monitoring Entitlements can be monitored by RHN Satellite. Monitoring includes system activity, availability. If defined



# Red Hat Network Client

- Registration
  - Run **rhnc\_register**
  - Select the updates location (RHN or local satellite/proxy)
  - Enter Account information
- Interactive usage
  - **yum** plug-in for downloading packages from RHN
  - Configuration in `/etc/yum/pluginconf.d/rhn-plugin.conf`
- Remote management
  - **rhnsd** polls RHN every four hours
  - **rhnc\_check** polls immediately

9-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**rhnc\_register** will create configuration files for **yum** as well as registering your system with RHN or your satellite-server.

RHN can be used to perform remote administration of collections of machines. First, actions for the machine (such as specific package installation or upgrades) are queued for the machine using the RHN account.

Client machines use the **rhnsd** daemon to poll RHN periodically for queued actions. By default, **rhnsd** polls every 4 hours, though this can be adjusted in `/etc/sysconfig/rhn/rhnsd`. The **rhnsd** daemon uses the `/usr/sbin/rhnc_check` command to actually perform the poll and administer any queued actions. Notably, the **rhnsd** daemon does not open any server networking ports.

## End of Unit 9

- Questions and Answers
- Summary
  - rpm -Uvh
  - rpm -q
  - yum
  - \*.repo files
  - How does Red Hat Network work?

9-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 9

## Working with packages

---

- Goal:** To gain working experience with package management
- System Setup:** A working install of Red Hat Enterprise Linux 5 connected to the classroom network
- Situation:** You have been asked to connect a system to your company's private **yum** repository to install and update software.

## Sequence 1: Using RPM

### Instructions:

1. Change to `/net/server1/var/ftp/pub`. In the Server directory, use **rpm -i** to install the `x3270-x11` RPM. This should fail. Correct the problem.
2. In the `errata` directory, use **rpm -i** to install the `autofs` RPM. This should fail. Correct the problem.
3. Use **rpm** queries to answer the following questions. In the blank spaces, write in the command used to find the answers.

What files are in the `initscripts` package?

rpm -ql

On what host was the `bash` RPM built, and what is its installed size?

hs20-bc2-2 5300518

Has the `pam` package changed since it was installed?

/etc/pam.d/system-auth

Which installed packages have "gnome" in their names?

Which RPM provides `/etc/inittab`?

initscripts

Which RPM provides `/etc/hosts`? Why?

? - to system

#### 4. RPM signatures

Practice checking the signature and integrity of an RPM package file of your choosing from your CD-ROM or from `server1`.

Import Red Hat's GPG key to RPM's system-wide keyring. The key can be found on first CD or `/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release`.

Check the signature of some original RPMs from the server.

Create a corrupted RPM, and then verify it. Start by copying an RPM file to `/tmp`, then use the **cat** command to append some extraneous data to the end of the file.

## Sequence 2: Connecting to a private repository

Scenario: You are asked to connect your system to the private repository located on `server1`.

Deliverable: A system configured to use the repository located on `server1`

Instructions:

1. Create a file `/etc/yum.repos.d/server1.repo` pointing to a repository with the name GLS located at the URL `http://server1.example.com/pub/gls/RPMS`. Make sure you enable the repository.

## Sequence 3: Installing new packages using yum

### Instructions:

1. Use yum to list all packages containing 'rhce-ts' in their name.
2. Install the package you just found in the previous step.

## Sequence 4: Updating software using yum

### Instructions:

1. Use **yum** to check if there are updates available for your system.

Replace your existing `/etc/yum.repos.d/server1.repo` file by downloading an updated copy from the URL `ftp://server1.example.com/pub/gls/server1.repo`. This will point yum to additional repositories containing the base Red Hat Enterprise Linux packages and available updates to those packages.

2. Select one package from the previous step and update it.

3. Now install all available updates for your system.

## Sequence 1 Solutions

1. Change to /net/server1/var/ftp/pub. In the Server directory, use **rpm -i** to install the x3270-x11 RPM. This should fail. Correct the problem.

```
# cd /net/server1/var/ftp/pub
# cd Server
# rpm -ivh x3270-x11*
warning: x3270-x11-3.3.4p73.el5.1.i386.rpm: Header V3 DSA ...
error: Failed dependencies:
    x3270 = 3.3.4p7 is needed by x3270-x11-3.3.4p73.el5.1.i386
```

The RPM is indicating it can not install until you resolve the dependencies. Install the x3270 RPM, then attempt x3270-x11 again.

```
# rpm -ivh x3270-3.3.4p7*
warning: x3270-3.3.4p73.el5.1.i386.rpm: Header V3 DSA ...
Preparing... ##### [100%]
1:x3270 ##### [100%]
# rpm -ivh x3270x11*
warning: x3270-x11-3.3.4p73.el5.1.i386.rpm: Header V3 DSA ...
Preparing... ##### [100%]
1:x3270-x11 ##### [100%]
```

2. In the errata directory, use **rpm -i** to install the autofs RPM. This should fail. Correct the problem.

- a. [root@stationX]# **cd ../errata**
- b. [root@stationX]# **rpm -ivh autofs\***  
warning: autofs...  
Preparing... ##### [100%]  
file /usr/lib/autofs/lookup\_file.so from install of  
autofs-5.0.10.rc2.43.0.2 conflicts with file from package  
autofs-5.0.10.rc2.42  
... output truncated ...
- c. The install failed, since another version of the RPM is already installed. This time, attempt an upgrade instead of an install.
- d. [root@stationX]# **rpm -Uvh autofs\***  
warning: autofs...  
Preparing... ##### [100%]  
1:autofs ##### [100%]

3. What files are in the initscripts package?

```
[root@stationX]# rpm -ql initscripts
```

On what host was the bash RPM built, and what is its installed size?



```
[root@stationX]# rpm -qi bash
```

Has the pam package changed since it was installed?

```
[root@stationX]# rpm -V pam
```

Which installed packages have "gnome" in their names?

```
[root@stationX]# rpm -qa | grep gnome
```

Which RPM provides /etc/inittab?

```
[root@stationX]# rpm -qf /etc/inittab
```

Using **rpm -qf** requires the full path to the file.

Which RPM provides /etc/hosts? Why?

```
[root@stationX]# rpm -qf /etc/fstab
```

No RPM provides /etc/hosts because this file is created by Anaconda during installation.

4. Import Red Hat's GPG key to RPM's system-wide keyring.

```
[root@stationX]# rpm --import   
/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Check the signature of an original RPM from the server.

```
[root@stationX]# cd /net/server1/var/ftp/pub/Server
```

```
[root@stationX]# rpm -K mutt-version.i386.rpm
```

Create a corrupted RPM, and then verify it. Start by copying an RPM file to /tmp, then use the **cat** command to append some extraneous data to the end of the file.

```
[root@stationX]# cp /net/Server1/var/ftp/pub/Server/mutt-version   
.i386.rpm /tmp
```

```
[root@stationX]# cat /bin/date >> /tmp/mutt-version.i386.rpm
```

```
[root@stationX]# rpm -K /tmp/mutt-version.i386.rpm
```

This command should fail.

## Sequence 2 Solutions

1. Create the file `/etc/yum.repos.d/server1.repo` with the following content:

```
[GLS]
name=Private classroom repository
baseurl=http://server1.example.com/pub/gls/RPMS
enabled=1
gpgcheck=0
```

Make sure you have configured the repository correctly by issuing the command: **yum list rhce-ts**.

## Sequence 3 Solutions

1. To list all packages containing 'rhce-ts' in their name you could issue the command: **yum list '\*rhce-ts'**
2. To install the package rhce-ts you could issue the command **yum install rhce-ts**  
When **yum** asks for confirmation enter **y**.

## Sequence 4 Solutions

1. Use **yum** to check if there are updates available for your system.

Replace your existing `/etc/yum.repos.d/server1.repo` file by downloading an updated copy from the URL `ftp://server1.example.com/pub/gls/server1.repo`. This will point yum to additional repositories containing the base Red Hat Enterprise Linux packages and available updates to those packages.

- a. `[root@stationX]# cd /etc/yum.repos.d`
- b. `[root@stationX]# mv server1.repo /tmp/`
- c. `[root@stationX]# wget ✓  
ftp://server1.example.com/pub/gls/server1.repo`
- d. To find out if there are updates available for your system use the command: **yum check-update**

2. Select one package from the previous step and update it.

- a. To update only a specific package you can use **yum update package-name**
- b. Install the kernel package:

```
[root@stationX]# yum update kernel
```

3. Now install all available updates for your system.

- a. To install all available updates for your system issue the command: **yum update**

# Unit 10

## System Administration Topics

10-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[ctraining@redhat.com](mailto:ctraining@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- configure X.org
- Configure printers
- Edit the system crontab
- System auto-mounter
- Understand PAM

10-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# XOrg Server Configuration

- Configured auto-magically at server start.
- Settings can be tuned:
  - Best results while in runlevel 3!
  - **system-config-display**
  - stored in `/etc/X11/xorg.conf`
- Requires the **xfs** service.
- Remote access via **ssh -X user@hostname**

10-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

`/etc/X11/xorg.conf` and specifies the hardware components' resources in several sections.

Among them are:

Server Layout

defining individual combinations of "InputDevice" and "Screen". Only one per session is used.

InputDevice

defining keyboard, mouse, touchscreen, or other form of supported input device.

Device

defining which hardware specific driver is used to communicate with the local video card.

Screen

defining individual combinations of "Monitor" and "Device" with display properties.

Please Note: X server configuration should be performed in runlevel 3 for best results. Also note that to run an X client to be displayed on a remote system, no local server configuration is necessary.

# CUPS

- uses the Internet Printing Protocol (IPP)
  - allows remote browsing of printer queues
  - based on HTTP/1.1
  - Uses PPD files to describe printers
  - Configuration files
    - `/etc/cups/cupsd.conf`
    - `/etc/cups/printers.conf`
  - Configuration tools
    - **system-config-printer**
    - Web based on `http://localhost:631`
    - Command line management with **lpadmin**

10-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Common Unix Printing System (CUPS) supports many advanced features such as printer pooling, automatic client configuration, job redirection and others. The Internet Printing Protocol (IPP) is based on HTTP/1.1. The web based management console of CUPS can be reached at port 631. By default only members of the `sys` group can connect. This console can also be used for configuration.

A mini-LPD service, **cups-lpd**, is available for backward compatibility with older LPRng client systems.

The main configuration file for `cupsd` is `/etc/cups/cupsd.conf`. This file is very similar in format to the Apache web server's `/etc/httpd/conf/httpd.conf` configuration file. Documentation in `/usr/share/doc/cups-version/` is also shared out as web pages through the web-based administrative interface. `/etc/cups/printers.conf` is automatically generated by the configuration tools and contains the configuration of the individual printer queues.

`system-config-printer` is a graphical utility that can ease setting up local and remote printers. It has the ability to set the system default printer and can configure CUPS to print to local and remote print queues including IPP, Windows, Novell and Unix LPD queues.

On the command line printer queues can be managed with **lpadmin**. CUPS has both a System V and BSD command line interface. You can use **lp lpr** to print. **lpstat** shows information about the local queues.



# System crontab Files

- Different format than user crontab files
- Master crontab file `/etc/crontab` runs executables in
  - `/etc/cron.hourly`
  - `/etc/cron.daily`
  - `/etc/cron.weekly`
  - `/etc/cron.monthly`
- `/etc/cron.d/` directory contains additional system crontab files

10-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The format of `/etc/crontab` and the files in `/etc/cron.d` are different from user crontabs. The sixth field is a username which will be used to execute the command in the seventh field.

A common command in these files is the run-parts shell script. This script takes one argument, a directory name, and invokes all of the programs in that directory (The run-parts script is located in `/usr/bin` and has no online documentation.).

The following is an example `/etc/crontab` file:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/bin:/usr/sbin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Thus, at 4:02 every morning, all of the executables in the `/etc/cron.daily` directory will be run as root. A default installation's `cron.daily` directory will contain scripts to update the `slocate` and `whatis` databases, to clean up temporary directories, and to perform other housekeeping tasks.

# Daily Cron Jobs

- **tmpwatch**
  - Cleans old files in specific directories
  - Keeps /tmp from filling up
- **logrotate**
  - Keeps log files from getting to large
  - Highly configurable in /etc/logrotate.conf
- **logwatch**
  - provides a summary about system activity
  - reports suspicious messages
  - Configuration file: /etc/logwatch/conf/logwatch.conf

10-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## tmpwatch

**tmpwatch** deletes all files in /tmp that have not been accessed for 240 hours (10 days). It also deletes files in /var/tmp that have not been accessed for 720 hours (30 days).

## logrotate

Left unchecked, system logs will grow until you run out of disk space. **logrotate** rotates log files from different subsystems at predefined intervals, or when they reach predefined sizes, and old logs are optionally compressed.

For example, /var/log/messages is rotated weekly to /var/log/messages.1, with older log files rotated to /var/log/messages.2, etc., optionally compressed.

Configuration is stored in /etc/logrotate.conf, for general settings, and /etc/logrotate.d/subsystem, for subsystem-specific settings. Service specific rotation rules are usually installed by the service's RPM.

## logwatch

Monitoring system logs is an onerous but important task. If you do not properly monitor your logs, you may miss security problems, hardware problems, or software problems. For example, a system may have a runaway maintenance problem every Monday at 5 AM that filled up the filesystem and then promptly cleaned it up again. Only the system log would reveal that there was a problem.

logwatch, installed by default on most Red Hat Enterprise Linux systems, monitors log files, reporting nightly on activity, and, potentially, on any anomalies located. logwatch is highly configurable. It can be

programmed to detect most any type of activity. See `/usr/share/doc/logwatch-version` for information on writing log filters.

# The anacron System

- **anacron** runs **cron** jobs that did not run when the computer is down
  - Assumes computers are not up continually
  - Vital for laptops, desktops, workstations, and other systems that are not up continually
  - Useful for servers that need to be taken down temporarily
- Configuration file: `/etc/anacrontab`
  - Field 1: If the job has not been run in this many days...
  - Field 2: wait this number of minutes after reboot and then run it
  - Field 3: job identifier
  - Field 4: the job to run

10-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **anacron** command now runs at boot time. The purpose of the **anacron** command is to run **cron** jobs that would have run had the computer not been down.

This is an essential tool for computers that are not constantly running, such as workstations, desktops, and laptop computers. It can also be useful for servers when they need to go down for an extended period of time. For example, the default configuration of **cron** runs the scripts in `/etc/cron.daily` once a day at 4:02am, according to the `/etc/crontab` file. Among the commands that are run from `cron.daily` is the **mlocate.cron** command, which recreates the **mlocate** database. If the laptop is almost always off at 4:02am, then the **mlocate** database will almost never be updated. For this reason, it is essential that **cron** jobs run on a regular basis. The solution for this is the **anacron** command.

Here is how **anacron** works:

When **cron** runs the **run-parts** command from `/etc/crontab` for `cron.daily`, `cron.weekly`, or `cron.monthly`, the first command to run is **0anacron**. This command sets a times tamp in a file in `/var/spool/anacron` that notes the time that this was last run.

On boot up, the **anacron** command runs. The `anacrontab` file specifies how often the commands in `cron.daily`, `cron.weekly`, and `cron.monthly` should run. If they have not run in this time, then **anacron** waits a few minutes (as specified in the second field of the `anacrontab` file) and then runs the commands, thus ensuring that if the computer was down during the time that **cron** should have run these commands, they are, nonetheless, run.

# Automounter

- System administrator specifies mount points controlled by automounter daemon process in `/etc/auto.master`
- The automounter monitors access to these directories and mounts the filesystem on demand
- Filesystems automatically unmounted after a specified interval of inactivity
- Enable the special map `-host` to "browse" all NFS exports on the network
- Supports wildcard directory names

10-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

By default, all mounted file systems are owned by the root account and can only be unmounted by the root account. This behavior can be overridden through the use of the `owner`, `user` or `users` options in `/etc/fstab`. This is normally done, for example, for removable media devices so that a non-privileged user can access the contents of a floppy disk or `cdrom`.

In addition to removable media devices, users may often need access to files located elsewhere on the network. This access can be obtained by mounting a network file share. Whether a network file share or some type of removable media device, one drawback is that casual users must learn the syntax of the mount and unmount commands. The automounter can be configured to monitor certain directories and automatically mount the appropriate devices when a reference is made to files in that directory. The automount daemon is provided by the `autofs` RPM.

To control filesystems with automount, modify the supplied `/etc/auto.master`.

The `/etc/rc.d/init.d/autofs` script parses this file and launches the automount daemon based on this configuration. Each line of `auto.master` lists a directory, present in the hierarchy, and a reference to yet another file that further defines specific mount options for those base mount points.

```
[root@stationX ~]# cat /etc/auto.master | grep "^/"
/misc    /etc/auto.misc
/net     -hosts
```

Referred from an `auto.master` entry, an `autofs` mount configuration file lists the filesystems to be mounted under the directory, including options required. Sample syntax can be found in the `autofs(5)` man page.

```
[root@stationX ~]# cat /etc/auto.misc | grep -v "#"
cd       -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
```

Once a mount point has been configured for automount, and the daemon started, merely requesting the filesystem (`ls`, `cd`, or other action) completes the mount. The daemon maintains the connection as long as the filesystem is in use plus an interval of time. The default interval is 60 seconds.

```
[root@stationX ~]# mount | grep cd
```

```
[root@stationX ~]# ls /misc
[root@stationX ~]# ls /misc/cd
autorun          README-ja.html          REALEASE-NOTES-gu.html
EULA             README-ko.html          REALEASE-NOTES-hi.html
... output truncated ...
[root@stationX ~]# mount | grep cd
/dev/sdc on /misc/cd type iso9660 (ro,nosuid,nodev)
```

Autofs mount configuration files support a wildcard system that allows multiple mounts to be specified within a directory. This is commonly used to mount home directories. If the following entry were in `/etc/auto.home.guests`, any directory within `server1:/home/guests` when the corresponding directory within `/home/guests` on the local system is accessed:

```
*      -fstype=nfs      server1:/home/guests/&
```

A special `-hosts` map can be configured in `/etc/auto.master` to browse NFS shares provided by other systems. The `-hosts` map only mount shares when accessed (lazy mounting), greatly reducing the load big file servers.

Browsing (also called ghosting) can be enabled globally in `/etc/sysconfig/autofs` by setting `DEFAULT_BROWSE_MODE="yes"` or by putting `-g` after the entry in `/etc/auto.master`.

# PAM Operation

- `/lib/security/` PAM modules
  - Each module performs a pass or fail test
  - Files in `/etc/security/` may affect how some modules perform their tests
- `/etc/pam.d/` PAM configuration
  - Service files determine how and when modules are used by particular programs

10-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

PAM modules are stored in `/lib/security/`. Each PAM module implements some test which may pass or fail. Some PAM modules may use supplementary configuration files in `/etc/security/` that control how they perform their tests. Others may use traditional files like `/etc/securetty` or may only be configurable through options.

The `/etc/pam.d/` directory contains service files. Each application that is PAM-aware has a service file which controls what modules it uses and how those modules affect the overall authentication results. Generally the service file that is used has the same name as the PAM-aware application. If the service file for an application is missing, `/etc/pam.d/other` is used as a default.

## /etc/pam.d/ Files: Tests

- Tests are organized into four groups:
  - `auth` authenticates that the user *is* the user
  - `account` authorizes the account may be used
  - `password` controls password changes
  - `session` opens, closes, and logs the session
- Each group is called as needed and provides a separate result to the service

10-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

PAM actually organizes tests into four management groups which are checked independently by different `libpam` library functions. The `auth` management group is used by PAM functions which authenticate users. The `account` management group is used to verify that an account is valid at this time and that passwords have not expired. The `password` management group is used to control password changes. The `session` management group is called by PAM at the start and at the end of a session.

Each management group returns independent results. A particular PAM function will only call PAM tests from one management group, but an application will generally perform tests from all management groups in turn. For example, on login it would be normal to perform `auth`, `account`, and `session` tests. If the password of an account had expired, it would not be unusual for the login program to prompt the user to change the password and perform `password` checks as well.

An individual PAM module may support all four management groups, but it does not need to. Some modules may be useful only for `auth` but not for `session`, or vice versa.



## /etc/pam.d/ Files: Control Values

- Control values determine how each test affects group's overall result
  - `required` must pass, keep testing even if fails
  - `requisite` as `required`, except stop testing on fail
  - `sufficient` if passing so far, return success now; if fails, ignore test and keep checking
  - `optional` whether test passes or fails is irrelevant
  - `include` returns the overall control value from tests configured in the file called

10-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

The second column of an `/etc/pam.d/` service file lists the control value that determines how the result of a module's test will affect the overall result of the management group. Modules are checked in order in the configuration file. A `required` module must pass for the overall result to pass. If it fails, the remaining modules are checked anyway to disguise why the failure happened from a potential attacker. A `requisite` module also must pass, but if it fails the failure is returned to the application immediately without checking other modules. This can be useful to stop a user from entering a password over an insecure terminal. If a `sufficient` module passes and the overall result so far is passing, then the pass is returned to the application immediately without checking other modules. If it fails, then the result is ignored. Whether an `optional` module passes or not has no effect on the return value.

Sometimes, a module will return *ignore* rather than pass or fail. This indicates that this test should be ignored when PAM figures out the overall return code for this management group.

An advanced control value syntax also exists for more complex scenarios. For instance,

```
[default=bad success=ok ignore=ignore user_unknown=ignore]
```

would work much like `required` except that errors due to the user not existing would be ignored rather than treated like a fail result. The advanced syntax is best avoided if possible.

## Important PAM Modules

- `pam_unix` - standard authentication
- `pam_env` - sets environment variables
- `pam_securetty` - limit root login to secure terminals
- `pam_stack` - calls another PAM service
- `pam_nologin` - tests for `/etc/nologin`
- `pam_deny` - always returns "failure" exit code
- `pam_cracklib` - password complexity checks
- `pam_console` - privileges for users at the console

10-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

...

## End of Unit 10

- Questions and Answers
- Summary
  - system-config-display
  - system-config-printer
  - /etc/crontab
  - autofs
  - PAM

10-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 10

## System Administration

---

Goal: Review basic system administration skills.

System Setup: Ensure the default system firewall is disabled.

```
# system-config-securitylevel
```

From the pull down menu, select **Disabled**.

## Sequence 1: CUPS printer administration.

**Scenario:** A printer has been installed to a system. It needs to be accessible locally and to other systems on the network. The system also needs to print to a remote printer on the network.

**Deliverable:** A system with a functioning local and remote print queue.

**Instructions:**

1. With **system-config-printer** create a local queue named `stationX-lp` with a description of "Local Printer on StationX". Select the connection of LPT #1, and use the generic, text-only, print driver.

*Note: Use the following for legacy-free PCs: Instead of selecting LPT #1, select LPD/LPR. From the **Hostname** pulldown, select `localhost`. From the **PrINTERname** pulldown, select `dummy`.*

2. Print a listing of `root`'s home directory. Since you do not have a printer attached, this job will be queued forever. Review the queue with **lpq -a**.
3. List the `/var/spool/cups` directory. You should a data file (d) and a control file (c). Display the data file. You probably don't want to **cat** the control file.
4. Launch **system-config-printer**. In the left hand pane, click **Server Settings**. Check the box marked "Share published printers connected to this system". Click Apply.
5. Add another printer to your system named `RemoteY-ipp` using the same drivers as the first. Rather than a local printer, make this an "Internet Printing Protocol (ipp)" device. For **Hostname**, use the IP address of a classmates system. For **PrINTERname**, use the name *they* specified.
6. Printer to the remote printer, and have a classmate print to your printer. List `/var/spool/cups` to see the new files queued.

*Note: Since there is no real printer, the queue may begin to reject incoming jobs. Issue **service cups restart** to force jobs to be accepted.*

## Sequence 1 Solutions

1. As root run **system-config-printer**. Select **New**. This will launch the installation druid. In the Printer Name field, input `stationX-lp`. For Description, input "Local Printer on StationX". Click Forward.

For Connection, select LPT #1, and click Forward.

*Note: Use the following for legacy-free PCs:* Instead of selecting LPT #1, select LPD/LPR. From the **Hostname** pulldown, select `localhost`. From the **PrINTERname** pulldown, select `dummy`, and click Forward..

By default, `Generic` will be highlighted. Accept this setting and click Forward. Under **Models**, select `text-only printer`;, and click Forward.

When presented with the screen entitled **Going to create a new printer...** select `Apply`. Close the application.

2. Print a listing of root's home directory. Since you do not have a printer attached, this job will be queued forever. Review the queue with **lpq -a**.

```
# ls /root | lpr -P stationX-lp
# lpq -a
```

3. List the `/var/spool/cups` directory. You should a data file (d) and a control file (c). Display the data file. You probably don't want to **cat** the control file.

```
# ls /var/spool/cups
c00001      d00001-001      tmp
# cat d00001-001
```

4. Launch **system-config-printer**. In the left hand pane, click **Server Settings**. Check the box marked "Share published printers connected to this system". Click `Apply`. Your printer is now shared to the other stations on the network.

5. Add another printer to your system named `RemoteY-ipp` by selecting **New Printer**. Rather than a local printer, make this an "Internet Printing Protocol (ipp)" device. For **Hostname**, use the IP address of a classmates system. For **PrINTERname**, use the name *they* specified.

Click Forward. `Generic` will be selected, click Forward again. In the **Models** pane, select "text-only printer", and click Forward. When presented with the screen entitled **Going to create a new printer...** select `Apply`. Close the application.

6. Printer to the remote printer, and have a classmate print to your printer. List `/var/spool/cups` to see the new files queued.

```
# ls /tmp | lpr -P RemoteY-ipp
# ls /var/spool/cups
c00001      c00002      d00001-001      d00002-001      tmp
```

*Note: Since there is no real printer, the queue may begin to reject incoming jobs. Issue **service cups restart** to force jobs to be accepted.*

# Unit 11

## Network Configuration

11-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.



# Objectives

Upon completion of this unit, you should be able to:

- Configure network interfaces
- Configure IPV6 networking
- Discuss remote access
- Configure network authentication
- Configure Xinetd
- Use network diagnostic tools

11-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Network Configuration Files

- /etc/sysconfig/network-scripts
  - Interface configuration
  - ifcfg-eth0
  - ifcfg-lo
- /etc/sysconfig
  - Global Network Parameters
  - network
- /etc
  - Name Resolution
  - hosts
  - resolv.conf

11-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Dynamic Configuration	Static Configuration
DEVICE=ethX HWADDR=0:02:8A:A6:30:45 BOOTPROTO=dhcp ONBOOT=yes Type=Ethernet	DEVICE=ethX HWADDR=0:02:8A:A6:30:45 IPADDR=192.168.0.254 NETMASK=255.255.255.0 GATEWAY=192.168.2.254 ONBOOT=yes Type=Ethernet

# Network Configuration Tools

- **system-config-network**
  - Device and Gateway
  - DNS and Hostname
- **system-config-network-tui**
  - Device and Gateway
- **Changes are not immediate**
  - **ifdown ethX, ifup ethX**
  - **service network restart**

11-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**ethtool**

**mii-tool**

# Address Types

## IP version 6

- Unicast
  - Single interface
  - Link-Local
    - Prefix - FE80::/10
    - Auto-configured for every interface to have one
    - Non-routable
  - Global
    - Routable
    - Dynamically or manually assigned to interface
- Reserved for Documentation
  - Prefix - 2001:0DB8::/32

11-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Address Types - part 2

### IP version 6

- Loopback - `::1/128`
  - Assigned to `lo` interface
- Multicast - `FF00::/8`
  - Set of interfaces - normally on different nodes
  - Delivered to "all members"
  - Broadcast obsoleted by this
- Anycast
  - Set of interfaces - normally on different nodes
  - Delivered by routers to "nearest" one
- Unspecified - `::/128`
  - Only used in software
- IPv4-mapped - `::FFFF:A.B.C.D/96`
  - Displayed by IPv6 services when bound to an IPv4 address

11-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Address Representation

## IP version 6

- 128 bits - what is best method to represent?
- Nibbles (four bits) - 32 hexadecimal digits
- Eight 16-bit (four nibble) segments separated by colons
  - 2001:0DB8:0000:0000:0216:41FF:FE59:0255
  - 2001:DB8:0:0:216:41FF:FE59:255 - leading zeros dropped
- Zero Compression
  - Strings of 0:0 replaced with ::
  - 2001:DB8:0:0:216:41FF:FE59:255
  - 2001:DB8::216:41FF:FE59:255
  - Can only be done "once" per address

11-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# New and Modified Utilities

## IP version 6

- **ping6**
- **traceroute6**
- **tracepath6**
- **ip -6**
- **host -t AAAA *hostname6.domain6***

11-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

- **/bin/ping6** - test connectivity and measure response time
- **/bin/traceroute6** - display the routers crossed to a remote host
- **/bin/tracepath6** - document the routers to a remote host and discover the MTU along this path
- **/sbin/ip** - view and/or configure IPv4 and IPv6 routing and device settings
- **/usr/bin/host** - lookup IPv6 address of host via DNS

# OpenSSH Overview

- OpenSSH replaces common, insecure network communication applications
- Provides user and token-based authentication
- Capable of tunneling insecure protocols through port forwarding
- System default configuration (client and server) resides in `/etc/ssh/`

11-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The OpenSSH project provides support for the secure shell protocol, a mechanism for providing secure authentication, remote execution, and remote login capabilities. In addition to the capabilities that the OpenSSH packages provide themselves, other packages, such as `rsync` and `rdist`, can use secure shell as their transport mechanism.

If a system does not need to provide remote shell access, but does need shell access to other hosts, then install the `openssh`, `openssl` and `openssh-clients` packages at a minimum, and add the `openssh-askpass` and `open-askpass-gnome` packages if you are running X. In order to provide remote ssh access to other systems, install the `openssh-server` RPM, which provides `sshd`. The `openssh-askpass*` packages are used in conjunction with **ssh-agent** in an X session.

Below is a list of the RPM packages and what they provide.

- `openssh`: **ssh-keygen**, **scp**
- `openssl`: cryptographic libraries and routines required by `openssh`
- `openssh-clients`: **ssh**, **slogin**, **ssh-agent**, **ssh-add**, **sftp**
- `openssh-server`: **sshd**
- `openssh-askpass`: X11 passphrase dialog
- `openssh-askpass-gnome`: GNOME passphrase dialog



# OpenSSH Server Configuration

- SSHD configuration file
  - `/etc/ssh/sshd_config`
- Options to consider
  - Protocol
  - ListenAddress
  - PermitRootLogin
  - Banner

11-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Several options are available to customize SSHD's operation. The configuration file (`/etc/ssh/sshd_config`) shipped with Red Hat Enterprise Linux has most of its entries "commented." These commented entries are the default configuration which might require modification to meet your security policy. Note, for example, that `X11Forwarding` has two entries. That commented is the software default. Red Hat installs a server that forwards X11 connections back to the connecting client.

SSHD is configured by default to listen for two protocols, SSH2(DSA) and the older SSH1(RSA). SSH1 has inherent security risks and is only provided for compatibility with older systems. It should be avoided whenever possible. SSH2 is the "preferred" protocol as configured by the "Protocol 2,1" option. To disable SSH1 protocol, change the option to read:

Protocol 2

By default SSHD is configured to listen on port 22. You can configure SSH to listen on multiple interfaces and multiple ports. This below configures a system to listen on tcp port 22 on a specific interface:

ListenAddress 192.168.0.250:22

By default root is allowed to login via **ssh**. To disable this feature, set "PermitRootLogin no". Some administrators want to disable logging in as root with a password, but allowing root to log in using public-key authentication. To enforce this policy, set the option "PermitRootLogin without-password". A malicious user will not know that the password option has been disabled for root.

If you set "PermitRootLogin forced-commands-only", root can execute commands on a remote system using public key authentication only.

It is a good idea to warn users of your policies as they make a connection to your system, perhaps that their connection is being logged. The example below would display the contents of `/etc/issue.net` when a connection is made, before authentication starts.



# VNC: Virtual Network Computing

- Allows to access or share a complete desktop over the network
- Uses significantly less bandwidth as pure remote X connections
- Server
  - Individual users can start a VNC server with the command: **vncserver**
  - Runs `$HOME/.vnc/xstartup` upon startup
  - Requires a VNC password which should not be identical to the system password
  - Servers can automatically be started via `/etc/init.d/vncserver`
- Client
  - connects to a remote VNC server with **vncviewer host:screen**
  - Unique screen numbers distinguish between multiple VNC servers running on the same host
  - supports tunneling through SSH: **vncviewer -via user@host localhost:1**

11-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Virtual Network Computing (VNC) is a platform independent way to access a complete desktop over the network. It uses significantly less resources than a traditional remote X11 connection. VNC sessions can be shared by multiple clients and also be used in “view-only” mode for demonstrations. In system administration VNC is most commonly used to administer machines graphically or to monitor installations.

Each user can setup a VNC password and start **vncserver** manually. The desktop is initialized using `$HOME/.vnc/xstartup`.

The system administrator can use `/etc/init.d/vncserver` to automate startup. This System V script reads `/etc/sysconfig/vncservers` and starts all VNC servers specified by `VNCSERVERS`.

Clients can access the server by using **vncviewer host:screen**. `host` is the hostname or IP of the remote server, `screen` the unique screen number assigned to the particular server process.

However this connection is not encrypted. Therefore it is strongly recommended to use the `-via user@host` option. This creates an SSH tunnel. Use `localhost` as the VNC host name in this case.

When a second VNC client connects to the same server the first one is disconnected. The first client can allow multiple connections by specifying the `-Shared` option.

# Authentication Configuration

- **system-config-authentication**
  - GUI tool to configure authentication
  - For text-based tool, use **authconfig-tui**
  - Load **authconfig-gtk** RPM
- **Supported account information services:**
  - (local files), NIS, LDAP, Hesiod, Winbind
- **Supported authentication mechanisms:**
  - (NSS), Kerberos, LDAP, SmartCard, SMB, Winbind

11-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The mechanisms used to manage user information can be changed with **system-config-authentication**. By default, this provides a graphical tool to configure network authentication. The interface provides two tabs: **User Information** (which changes NSS settings) and **Authentication** (which changes PAM settings). Check boxes are used to select which user information sources or authentication mechanisms are desired, and then each service must be configured through button selections. In addition, use of shadow passwords or the MD5 password encryption algorithm can be toggled on or off. The **--nox** option will open this tool as a text-based menu interface instead, usable without an X display. In that case, the first screen will allow selection of the user information sources or authentication mechanisms, and subsequent screens will be used to configure those interfaces.

For **User Information** five data sources are supported:

If nothing is selected, only local files will be used as a source of NSS information. NIS gets information from database maps stored on a NIS server. LDAP allows account information to be stored as entries in a LDAP directory server. Hesiod stores user information as special resource records in a DNS name server, and its use is relatively uncommon. Winbind uses winbindd to automatically map accounts stored in a Microsoft Windows domain controller to Linux users by storing SID to UID/GID mappings in a database and automatically generating any other NSS information that is required.

For **Authentication**, six data sources are supported:

If nothing is selected, it is assumed that NSS will provide an encrypted password with the other NSS user information that can be compared to the entered password normally. Kerberos authenticates users by requesting a *ticket* for the user from the Kerberos server, and if the user's password decrypts the ticket the authentication passes. LDAP authentication maps the username provided to a LDAP directory entry and tries to bind to the directory using that entry and the provided password; if this succeeds, the authentication passes. SmartCard authentication allows a SmartCard to be used to login. Once removed, it can also be used to lock the system. SMB and Winbind use different approaches to authenticate using a Microsoft Windows® domain controller.

The older version of this tool in Red Hat Enterprise Linux 3 and earlier was **redhat-config-authentication**, and it had a slightly different look and feel. The **authconfig** command can

be called as **authconfig-gtk** or **authconfig-tui** for graphical or text-based utility respectively, and also supports a **--kickstart** option which can make these settings through command-line flags.

## Example: NIS Configuration

- Must install **ypbind** and **portmap** RPMs
- Run **system-config-authentication**
  - Enable NIS to provide User Information
  - Specify NIS server and NIS domain name
  - Keep default authentication (through NSS)
- What does this actually do?
  - Five text-based configuration files are changed

11-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

One popular service used to centrally manage system and account information is NIS. NIS uses one or more NIS servers, each running **ypserv**, to share information with NIS client systems, running **ypbind**. The master server may also run **rpc.yppasswdd**, which allows users on NIS clients to update the passwords stored in NIS. Both NIS clients and NIS servers also must run a local service called **portmap** which helps remote systems contact the local **ypserv** or **ypbind** program. Clients and servers which communicate with each other are normally members of the same NIS domain, identified by an arbitrary name.

NIS servers typically are used to synchronize account information. They can share the contents of the `/etc/passwd`, `/etc/shadow`, `/etc/group`, and `/etc/gpasswd` files by converting them into *NIS maps*. Each NIS map consists of a set of key/value pairs. For instance, one typical NIS map used is `passwd.byname`, where the key is a username and the value is the matching line of user information for that account in `/etc/passwd` format.

The easiest way to set up a client to use an existing NIS server is to install the **portmap** and **ypbind** packages (which are part of the base install), and run **system-config-authentication**. Under **User Information**, enable NIS, and then a NIS domain and a NIS server for that domain must be specified. No changes are necessary under **Authentication** if NIS will be used for authentication, since it provides password hash information through NSS.

What does this change? The variable `NISDOMAIN` is set in `/etc/sysconfig/network` to the NIS domain's name, and the `domainname` command is run to set it. The `/etc/yp.conf` file has a line added to it which specifies which server to use for that NIS domain. The `/etc/nsswitch.conf` file is modified to specify that NIS should be used as a source of information for password, shadow, group, and other lookups. The `USENIS=yes` variable is set in `/etc/sysconfig/authconfig`. Finally, `/etc/pam.d/system-auth-ac` is modified so that password change requests for NIS accounts will be sent to the **rpc.yppasswdd** service running on the NIS master server.

NIS is a relatively insecure service. Kerberos authentication can be used in conjunction with NIS to improve password security. A better solution might use LDAP protected with TLS (SSL) encryption to store name service information in place of NIS. However, these solutions can be more complex to set up and manage.

## Example: LDAP Configuration

- Must install `nss-ldap` and `openldap` RPMs
- Run **system-config-authentication**
  - Enable LDAP to provide User Information
  - Specify server, the search base DN, and TLS
  - Enable LDAP to provide Authentication
- What does this actually do?
  - Five text-based configuration files are changed

11-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

LDAP is a protocol used to talk to a distributed directory service based on X.500, which can be used to store system and account information. Client programs can use the LDAP protocol to get information from directory servers running **slapd**, the standalone LDAP service. The OpenLDAP packages in the distribution provide a **slapd** server implementation as well as client tools and development libraries which can be used to work with LDAP services. Information is stored in the directory in individual entries organized into a hierarchical tree. Each entry and its location in the tree is uniquely identified by its *distinguished name*, or DN. A particular LDAP server is normally responsible for only the part of the tree under a particular entry, and the DN of this entry is normally used as a base DN for searches when looking up information stored in that server. It may help to think of this base DN like a DNS domain name, and the entire LDAP directory tree like all of DNS.

One use of an LDAP directory service is to synchronize account information between multiple networked systems. An individual entry may represent a single user (by using information from the relevant lines in `/etc/passwd` and `/etc/shadow`), or a single group (using information from the relevant line in `/etc/group`), or may store other information.

The easiest way to set up a client to use an existing LDAP server is to install the `nss_ldap` and `openldap` packages, and run **system-config-authentication**. To get NSS information from LDAP, under **User Information**, enable LDAP, and then specify an LDAP server, and a base DN to use for searches. If the LDAP service provides standard password hashes to NSS, this may be sufficient. Alternatively, under **Authentication** you can also enable LDAP, which will let PAM test and change passwords by accessing the directory service using the DN of the account's entry and the password entered. If you select LDAP on the **Authentication** tab, it is crucial to also check **Use TLS to encrypt connections**, or your unencrypted password will be transmitted over the network to the LDAP server as clear-text on every authentication!

What do these settings change? The `/etc/ldap.conf` file is modified to specify the location of your LDAP server, your search base DN, and whether or not TLS is enabled. The `/etc/openldap/ldap.conf` file is modified with the same information so that command-line OpenLDAP tools and the automounter can use the same server and base DN. The `/etc/nsswitch.conf` file is modified to specify that LDAP should be used as a source of information for password, shadow, group, and other lookups. The `USELDAPAUTH=yes`

and/or USELDAP=yes variables are set in `/etc/sysconfig/authconfig`. Finally,  
`/etc/pam.d/system-auth-ac` is modified so that PAM will try accessing the directory service as  
your account entry to authenticate access.



# The xinetd service

- Manages transient services upon demand
  - less-frequently needed services
  - host-based authentication
  - service statistics and logging
  - service IP redirection
- Configuration files: `/etc/xinetd.conf`,  
`/etc/xinetd.d/service`

11-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Services which are needed less frequently, or requiring additional resource management, are typically controlled by the **xinetd** service. This daemon provides host-based authentication, resource logging, timed access, and address redirection among its many configuration options. These options may be service-specific, or generally applied across all **xinetd**-managed services. **xinetd** uses `/etc/services` in its configuration of port-to-service management.

The default installed configuration of **xinetd** is provided by the top-level configuration file `/etc/xinetd.conf` and service specific files under the `/etc/xinetd.d` directory tree.

The top-level configuration file `/etc/xinetd.conf` sets the global configuration options shared by all managed services. It also provides the path to service specific configurations. Below is an annotated version of the default installed top-level configuration file.

The `includedir /etc/xinetd.d` directive includes each file in `/etc/xinetd.d/`. Each individual service can override the global settings in `/etc/xinetd.conf`.

More information may be found in the `xinetd-*` directory under `/usr/share/doc`, and `man 5 xinetd.conf`.

# xinetd service controls

- Service specific configuration

- `/etc/xinetd.d/service`

```
/etc/xinetd.d/tftp:
# default: off
service tftp
{
    disable = yes
    server  = /usr/sbin/in.tftpd
}
```

11-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

Above is a portion of the default service specific configuration file for `tftp`. Perhaps the two most important directives are `disable` and `server`.

The `disable` line determines whether or not **xinetd** will accept connections for the service. If this parameter is not present, the default is "not disabled". Most service are installed `disable = yes`. For the service to become active, either the file needs to be edited, and `xinetd` restarted, or, a utility such as **chkconfig** must be used.

```
# grep "disable" /etc/xinetd.d/tftp
    disable = yes
# chkconfig tftp on
# grep "disable" /etc/xinetd.d/tftp
    disable = no
```

The `server` entry defines the binary used to run the service.

# Network Diagnostic Tools

- nmap - port scanner
- tcpdump - packet monitor
- wireshark (tshark) - packet "sniffer"

11-17

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2894 or +1 (919) 754 3700.

# End of Unit 11

- Questions and Answers
- Summary
  - system-config-network
  - IPV6
  - SSH, VNC
  - system-config-authentication
  - xinetd
  - tcpdump, nmap, wireshark

11-18

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 11

## Network

---

Goal: Understand IPv4 and IPV6 integration in Red Hat Enterprise Linux.

System Setup: A system with access to the `example.com` network.

## Sequence 1: Using IPv6

Scenario: A network is migrating to IPv6 and you need to test some connectivity issues.

Deliverable: The ability to do basic IPv6 network exploration.

### Instructions:

1. Use **ifconfig eth0** to review your network settings. Notice your ethernet card's HWaddr and your IPv6 address. Do you see some common characters?
2. Notice your IPv6 Scope. Ping your IPv4 address with a count of one. Use **ping6** to ping your IPv4 address with a count of one. This should fail. Check **man ping6** for the **-I** option. Do you understand why it failed? Make the ping work.
3. Use SSH to connect to your IPv6 localhost address. This should work. Use SSH to connect to your IPv6 link-local address. This should fail for the same reason **ping6** failed on the first attempt.
4. Unfortunately, **ssh -I** does not work the same way as with ping. Keep in mind that there are two different ping commands, but SSH is trying to use both networking stacks at the same time. Reissue the command, appending **%eth0** to the address.
5. Obviously, these address are difficult to remember, unlike IPv4 address. Ultimately, the IPv6 addresses will be managed by DNS. Ping your own hostname as follows:

```
ping6 -c1 -I eth0 stationX.example.com
```

This should fail. Try **server1.example.com**. This should work. What is the difference? Use the **host** to find out.

## Sequence 2: Exploring Xinetd Services

**Scenario:** A system needs a Telnet package installed which is managed by the Xinetd super-daemon. The service needs to be available on demand, but no Telnet processes should exist on the system when users are not connected.

**Deliverable:** A functioning Telnet server.

**Instructions:**

1. Attempt to telnet to localhost. This will fail. Since we know that telnet is a Xinetd related service, check the /etc/xinetd.d directory for a telnet file. There should not be a "telnet" file, as Red Hat Enterprise Linux does not install a telnet server by default.
2. Install telnet-server. Look very closely at what was installed. You will notice that not only was telnet-server installed, but so was xinetd. On earlier versions of Red Hat Enterprise Linux, xinetd was a base install item. Ensure the Xinetd service is active and will start at boot time.
3. Again, attempt to telnet to localhost. This should also fail. Install nmap and use it to scan localhost. Why isn't the telnet port open?
4. To get telnet-server to answer, disable=yes needs to change to disable=no. One possibility is to edit the file, but we would then need to reload the Xinetd service for the daemon to recognize the change. Another option is to use chkconfig to change the file and reload Xinetd.
5. At this point telnet is enabled, but it is not active. Execute the following:

```
# xterm -e "watch 'ps x | grep 0.[itx]' " &
```

This will launch a monitoring window that will let us keep track of telnet's processes. (Notice that line ends with both a single quote and a double quote.) Use telnet to connect to localhost. Watch for any changes in the monitoring window.

6. Have a classmate telnet to your system and observe the changes in the monitoring window.

## Sequence 3: Client-side NIS account management

**Scenario:** Your site is centrally managing user account information in a NIS directory service on 192.168.0.254, using the NIS domain name `notexample.e`. Set up your client to get user information from NIS, authenticating users with the NSS information provided by the NIS server. In addition, home directories for these users are exported through NFS to all workstations as well. You should set up the automounter to automatically mount and unmount these directories as needed.

**Deliverable:** Your system configured as an NIS client.

### Instructions:

1. Install the necessary RPMs on your system. By default, `portmap` is installed. You will also need: `ypbind`, `yp-tools`, `authconfig`, and `authconfig-gtk`.
2. Configure your client to use NIS for authentication. Use `notexample.e` as the domain and `192.168.0.254` as the server.
3. Restart the `sshd` service to make sure that it registers the changes to authentication. You can test NIS with the user `guest20XX` (where `XX` is a two-digit version of your station number), and a password of `password`. This should succeed, but display an error message about the missing home directory.
4. Use the automounter to mount the home directories for your NIS users from `server1.example.com`. You can use `getent passwd` to see what home directories are assigned. `server1.example.com` exports `/home/guests` to your system.
5. Once you have successfully connected via NIS, use `system-config-authentication` to disable NIS before moving on to the next exercise. There is no reason that NIS and LDAP can not co-exist on the same system, but we will disable the service to ease troubleshooting of the next exercise.



## Sequence 4: Client-side LDAP account management

**Scenario:** Your site is migrating new user account information into an LDAP directory service, also served by `server1.example.com` under the search base `dc=example,dc=com`. The administrator of the LDAP server requires clients to use TLS (SSL) encryption, and the LDAP server's TLS certificate is digitally signed by a locally-run TLS Certificate Authority so that clients may verify that they are communicating with the real LDAP server.

Using TLS encryption, set up your client system to get user information and authenticate users using the LDAP server. Home directories for these LDAP-managed users use the same NFS export as your old NIS-managed users, so you will not have to make any changes to your automounter configuration.

**Deliverable:** Your system configured as an LDAP client.

### Instructions:

1. Install the necessary RPMs on your system. Your system should already have the `authconfig` and `authconfig-gtk`. Several RPMs are needed for LDAP: `openldap`, `openldap-clients`, and `nss_ldap`.
2. Use **system-config-authentication** to configure your client to use LDAP for user information and authentication. The LDAP server is `server1.example.com`. The Search Base DN is: `dc=example,dc=com`. The server is only available via TLS. You will need to *Download CA Certificate* from the following URL:

`ftp://server1/pub/EXAMPLE-CA-CERT`

3. Restart the **sshd** service to make sure that it registers the changes to authentication. You can use the user account `ldapuserX` with a password of `password` for testing. This should succeed. Why didn't you get a missing directory error?
4. If you see any errors, look at the logs in `/var/log/messages` and `/var/log/secure`. You can also try running the command **ldapsearch -x -Z**, which should dump user information in LDIF format from your LDAP server to standard output if the server is reachable.

TLS can be tested with **openssl s\_client**. See `s_client(1)` for details.

5. **MANDATORY CLEANUP**

Once you have successfully authenticated via LDAP, use `system-config-authentication`, to disable the feature. This will prevent unnecessary network traffic from interfering with future exercises.

## Sequence 1 Solutions

1. Use **ifconfig eth0** to review your network settings. Notice your ethernet card's HWaddr and your IPv6 address. Do you see some common characters?

```
# ifconfig eth0
Link encap:Ethernet HWaddr 00:0D:60:FA:F5:F2
inet addr:192.168.0.X Mask:255.255.255.0
inet6 addr: fe80::20d:60ff:fefa:f5f2/64 Scope:Link
... output truncated ...
```

In the example, look at the last two sets of hex values in the MAC address: F5:F2. Look at the last segment of the IPv6 address: f5f2. Look at the middle two sets of hex values in the MAC address: 60:FA. Look at the middle two segments of the IPv6 address: 60ff:fefa. Compare the value in the example to the middle two segments of your IPv6 address. You should notice that the both contain ??ff:fe??. As you can see, this number is unique to your ethernet card.

2. Notice your IPv6 Scope. Ping your IPv4 address with a count of one. Use **ping6** to ping your IPv4 address with a count of one. This should fail. Check **man ping6** for the -I option. Do you understand why it failed? Make the ping work.

```
# ping -c1 192.168.0.X
PING 192.168.0.X (192.168.0.X) 56(84) bytes of data.
64 bytes from 192.168.0.X: icmp_seq=1 ttl=64 time=0.056 ms
... output truncated ...
# ping6 -c1 fe80::20d:60ff:fefa:f5f2
connect: Invalid argument
```

The man page tells us to use the -I ethX option when ever we are Scope:Link. At issue is that all of our network connections could be Scope:Link, so **ping6** does not know which one to use. By specifying the interface, our system is not broadcasting, thus generating needless traffic.

```
# ping6 -c1 fe80::20d:60ff:fefa:f5f2 -I eth0
PING fe80::20d:60ff:fefa:f5f2 from fe80::20d:60ff:fefa:f5f2 ↵
eth0: 56 data bytes
64 bytes from fe80::20d:60ff:fefa:f5f2: icmp_seq=0 ttl=64 ↵
time=0.134 ms
... output truncated ...
```

3. Use SSH to connect to your IPv6 localhost address. This should work.

```
# ssh ::1
... output truncated ...
root@::1's password:
```

Use SSH to connect to your IPv6 link-local address. This should fail for the same reason **ping6** failed on the first attempt.

```
# ssh fe80::20d:60ff:fefa:f5f2
ssh: connect to host fe80::20d:60ff:fefa:f5f2 port 22: Invalid ↵
```

argument

4. Unfortunately, **ssh -I** does not work the same way as with ping. Keep in mind that there are two different ping commands, but SSH is trying to use both networking stacks at the same time. Reissue the command, appending `%eth0` to the address.

```
# ssh fe80::20d:60ff:fefa:f5f2%eth0
... output truncated ...
root@fe80::20d:60ff:fefa:f5f2%eth0's password:
```

5. Obviously, these address are difficult to remember, unlike IPv4 address. Ultimately, the IPv6 addresses will be managed by DNS. Ping your own hostname as follows:

```
# ping6 -c1 -I eth0 stationX.example.com
unknown host
```

This should fail. Try `server1.example.com`.

```
ping6 -c1 -I eth0 server1.example.com
PING server1.example.com(fe80::250:bfff:fe19:7a07) from
fe80::20d:60ff:fefa:f5f2 eth0: 56 data bytes
64 bytes from fe80::250:bfff:fe19:7a07: icmp_seq=0 ttl=64
time=3.84 ms
```

This should work. What is the difference? Use the **host** to find out.

```
# host -t AAAA stationX.example.com
stationX.example.com has no AAAA record
# host -t AAAA server1.example.com
server1.example.com has IPv6 address fe80::250:bfff:fe19:7a07
```

The instructor has made an entry in the DNS server for the his system, but not the student workstations. Either your instructor is overworked or lazy.

## Sequence 2 Solutions

1. Attempt to telnet to localhost. This will fail.

```
# telnet localhost
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
telnet: Unable to connect to remote host: Connection refused
```

Since we know that telnet is a Xinetd related service, check the /etc/xinetd.d directory for a telnet file. There should not be a "telnet" file, as Red Hat Enterprise Linux does not install a telnet server by default.

```
# ls /etc/xinetd.d/telnet*
ls: /etc/xinetd.d/telnet*: No such file or directory
# rpm -qa telnet-server
```

- 2.

```
# yum install -y telnet-server
... output truncated ...
Installed: telnet-server.x86_64 1:0.17-38.el5
Dependency Installed: xinetd.x86_64 2:2.3.14-10.el5
Complete!
# service xinetd status
xinetd is stopped
# service xinetd start
Starting xinetd: [ OK ]
# chkconfig xinetd --list
xinetd    0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

3. Again, attempt to telnet to localhost. This should also fail.

```
# telnet localhost
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
telnet: Unable to connect to remote host: Connection refused
```

Install nmap and use it to scan localhost.

```
# yum install -y nmap
... output truncated ...
Installed: nmap.x86_64 2:4.11-1.1
Complete!
# nmap localhost
... output truncated ...
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
```

The telnet port is not open, because Xinetd services are install disabled by default.

```
# grep disable /etc/xinetd.d/*  
... output truncated ...  
/etc/xinetd.d/telnet:  disable    = yes
```

4. To get telnet-server to answer, disable=yes needs to change to disable=no. One possibility is to edit the file, but we would then need to reload the Xinetd service for the daemon to recognize the change. Another option is to use chkconfig to change the file and reload Xinetd.

```
# chkconfig telnet on  
[root@stationX ~]# grep disable /etc/xinetd.d/telnet  
disable = no
```

5. At this point telnet is enabled, but it is not active. Execute the following:

```
# xterm -e "watch 'ps x | grep 0.[itx]' " &
```

This will launch a monitoring window that will let us keep track of telnet's processes. (Notice that line ends with both a single quote and a double quote.) Initially you will see few entries, including:

```
1853 ?  Ss  0.00  xinetd -stayalive -pidfile /var/run/xinetd.pid
```

This is the xinetd daemon. Use telnet to connect to localhost.

```
# telnet localhost  
... output truncated ...  
login:
```

Notice the monitoring window. This time you will entries similar to the following

```
3568 pts/0  S+  0:00 telnet localhost  
3569 ?      Ss   0:00 in.telnetd: stationX
```

6. Have a classmate telnet to your system and observe the changes in the monitoring window.

## Sequence 3 Solutions

1. Install the necessary RPMs on your system if they are not already installed. For example, to test for portmap, you can run **rpm -q portmap**. Use yum to install the others:

```
# yum install -y ypbind yp-tools authconfig authconfig-gtk
```

2. Configure your client to use NIS for authentication. Use notexample as the domain and 192.168.0.254 as the server.

If you are using the graphical interface, check the option **Enable NIS Support** on the **User Information** tab. You can leave the **Authentication** tab alone. Click the **Configure NIS...** button, and in the **NIS Settings** window specify notexample for the **NIS Domain** and server1.example.com for the **NIS Server**. Click on the **OK** button for the **NIS Settings** window, and again on the main window.

If you are using a (text-based) virtual console or the **authconfig-tui** command, then under **User Information** check **Use NIS**. Under **Authentication** leave **Use MD5 Passwords** and **Use Shadow Passwords** checked. Then select the **Next** button, and on the next screen specify notexample for **Domain** and 192.168.0.254 for **Server**. Select **Ok**.

Either way, ypbind should now start up successfully. If it does not, check the configuration for misspellings, or /var/log/messages for errors.

3. Restart the **sshd** service to make sure that it registers the changes to authentication.

```
# service sshd restart
```

You can test NIS with the userguest20XX (where XX is a two-digit version of your station number), and a password of password.

```
# ssh guest20XX@stationX.example.com
```

The **ssh** command should succeed, but display an error message about the missing home directory. We will fix this in the next step.

If the password was not accepted, look at the log files. Do you see an SELinux error? Double check your settings. Did you use the hostname for server1 rather than the IP address?

4. Use the automounter to mount the home directories for your NIS users from server1.example.com. You can use **getent passwd** to see what home directories are assigned. server1.example.com exports /home/guests to your system.

Begin by editing /etc/auto.master and add the following line:

```
/home/guests /etc/auto.guests --timeout=60
```

This line specifies that /etc/auto.guests defines mount points in /home/guests managed by the automounter. When not in use for more than 60 seconds, filesystems mounted on those mount points are automatically unmounted.

Create and edit /etc/auto.guests so it contains the following:

```
* -rw,soft,intr 192.168.0.254:/home/guests/&
```

This line specifies that access to any immediate subdirectory of /home/guests should make autofs mount a NFS export from 192.168.0.254 where the & is the same as the name of the local subdirectory. (So the automounter would mount 192.168.0.254:/home/guests/guest2001 on /home/guests/guest2001). The middle column specifies the mount options that will be used; read-write, timeout eventually if the NFS server is not available, and timeout immediately if an interrupt is sent.

Configure **autofs** to start in run levels 2, 3, 4, and 5, then restart it manually:

```
# chkconfig autofs on ; service autofs restart
```

Now try logging in again and see whether the home directory gets mounted automatically. #It should. Try logging into to your neighbors system once it is also configured. You should be able to access your home environment from any system in the notexample domain.

5. Once you have successfully connected via NIS, use system-config-authentication to disable NIS before moving on to the next exercise. There is no reason that NIS and LDAP can not co-exist on the same system, but we will disable the service to ease troubleshooting of the next exercise.

## Sequence 4 Solutions

1. Install the necessary RPMs on your system. Your system should already have the `authconfig` and `authconfig-gtk`. Several RPMs are needed for LDAP: `openldap`, `openldap-clients`, and `nss_ldap`.

```
# yum install -y openldap openldap-clients nss_ldap
```

2. Use **system-config-authentication** to configure your client to use LDAP for user information and authentication. In the graphical interface, on the **User Information** tab check **Enable LDAP Support**. Switch to the **Authentication** tab and also check **Enable LDAP Support**.

Select the **Configure LDAP...** button on either tab. On the window that opens, set your **LDAP Search Base DN** to `dc=example,dc=com`. Set your **LDAP Server** to `server1.example.com`.

Using TLS is critical for security if you enable LDAP on the Authentication tab, because the PAM module used for LDAP will send passwords as clear-text over the network to the directory server on each authentication if this is not selected. Finally, press the **Download CA Certificate** button and enter the following in the **Certificate URL**:

```
ftp://server1/pub/EXAMPLE-CA-CERT
```

*Note:* If you do not add the certificate, you will notice the warning box about the missing certificate file.

3. Restart the `sshd` service to make sure that it registers the changes to authentication. You can use the user account `ldapuserX` with a password of `password` for testing.


```
# service sshd restart
```

```
# ssh ldapuserX@stationX.example.com
```

This should succeed. In the NIS exercise, we saw a missing directory error. Since the LDAP users are also assigned under `server1`'s `/home/guests` directory, they are using the same automount configuration as the NIS users.

4. If you see any errors, look at the logs in `/var/log/messages` and `/var/log/secure`. You can also try running the command `ldapsearch -x -Z`, which should dump user information in LDIF format from your LDAP server to standard output if the server is reachable.

TLS can be tested with `openssl s_client`. See `s_client(1)` for details.

```
a. [root@stationX]# openssl s_client -connect 
server1.example.com:443
```

5. MANDATORY CLEANUP



Once you have successfully authenticated via LDAP, use system-config-authentication, to disable the feature. This will prevent unnecessary network traffic from interfering with future exercises.

# Unit 12

## Network Security

12-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Understand tcp\_wrappers
- Recognize applications using libwrap.so
- Understand iptables firewalling
- Customize iptables firewall rules

12-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# tcp\_wrappers Configuration

- Three stages of access checking
  - Is access explicitly permitted?
  - Otherwise, is access explicitly denied?
  - Otherwise, by default, permit access!
- Configuration stored in two files:
  - Permissions in `/etc/hosts.allow`
  - Denials in `/etc/hosts.deny`
- Basic syntax:

```
daemon_list: client_list [:options]
```

12-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

`/etc/hosts.allow` and `/etc/hosts.deny` each have two or more colon-separated fields. The first field specifies a comma-delimited list of executable names (*not* service names), possibly with the wildcards `ALL` and `EXCEPT`. The second field contains a comma-separated list of client specifications, using IP address, hostname, "trailing dot" networks, "leading dot" domains, or network/netmask pairs. Again, the keywords `ALL` and `EXCEPT` are recognized.

When parsing the files, `libwrap.so` implements a "stop on first match" policy: as soon as a daemon/client configuration line is matched, the configuration line is implemented, and then no further action occurs. A matching line in `/etc/hosts.allow` would allow the connection. A matching line in `/etc/hosts.deny` would deny the connection. First, `/etc/hosts.allow` is examined. If access is not explicitly allowed, `/etc/hosts.deny` is examined. If access is not explicitly denied, the connection is allowed, *by fault of omission*: that is, the connection request meets *no* rule criterion.

Changes to the access files are effective immediately for all new connections.

# Daemon Specification

- **Daemon name:**
  - Applications pass name of their executable
  - Multiple services can be specified
  - Use wildcard `ALL` to match all daemons
  - Limitations exist for certain daemons
- **Advanced Syntax:**

`daemon@host: client_list ...`

12-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The first field specifies a comma-delimited list of daemons. `tcp_wrappers` usually takes `argv[0]` (the name of the process without path) as the daemon name.

Examples:

```
in.telnetd: 192.168.0.1
sshd, gdm: 192.168.0.1
```

If your host has more than one network interface and you want to implement different policies for them, use the following syntax:

```
in.telnetd@192.168.0.254: 192.168.0.
in.telnetd@192.168.1.254: 192.168.1.
```

To block access to RPC-based services like NFS or NIS, block the underlying **portmap**. Unlike **xinetd**-managed services or **gdm**, for which changes to the access control lists take place immediately, it takes a brief interval of time for changes to rules concerning portmap to take effect. Remember that both NFS and NIS rely on the **portmap** daemon.

# Client Specification

- Host specification
  - by IP address (192.168.0.1, 10.0.)
  - by name (www.redhat.com, .example.com)
  - by netmask (192.168.0.0/255.255.255.0)
  - by network name

12-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Client Specification

- by IP-address
  - Full or partial IP addresses.
  - Rightmost components are treated as zero if omitted
  - Example: 192.168.1. (all hosts within the class C network 192.168.1.0)
- by network / netmask
  - Specify the complete network address plus netmask.
  - Netmask must be in the long format.
  - Example: 192.168.0.0/255.255.255.0
- by host name
  - Performs a reverse lookup every time a client connects.
  - Is not always supported.
  - Example: .example.com (all hosts in the example.com domain)
- by network name
  - network names from /etc/networks or NIS.
  - Does not work together with usernames.
  - Example: @mynetwork

# Advanced Client Syntax

- Wildcards
  - ALL, LOCAL
  - KNOWN, UNKNOWN, PARANOID
- EXCEPT operator
  - Can be used for client and service list
  - Can be nested

12-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## *Wildcards*

ALL always matches

LOCAL all hosts without a dot in their name

UNKNOWN all hosts or users that cannot be looked up

KNOWN all hosts or users that can be determined

PARANOID matches all hosts where lookup and reverse lookup do not match

## *EXCEPT Operator*

The EXCEPT operator can be used in daemon and client lists to exclude some hosts from your match. It can be nested. For example, consider the following:

```
/etc/hosts.allow
sshd: ALL EXCEPT .cracker.org EXCEPT trusted.cracker.org
```

```
/etc/hosts.deny
sshd: ALL
```

Because of the catch-all rule in `hosts.deny` this rule set would allow only those who have been explicitly granted access to `ssh` into the system. In `hosts.allow` we grant access to everyone except for hosts in the `cracker.org` domain, but to this rule we make an exception: We will allow the host `trusted.cracker.org` to `ssh` in despite the ban on `cracker.org`.

## tcp\_wrappers Example

```
# /etc/hosts.allow
ALL : 127. [::1]
vsftpd : 192.168.0.
in.telnetd, sshd : .example.com 192.168.2.5

# /etc/hosts.deny
ALL : ALL
```

12-7

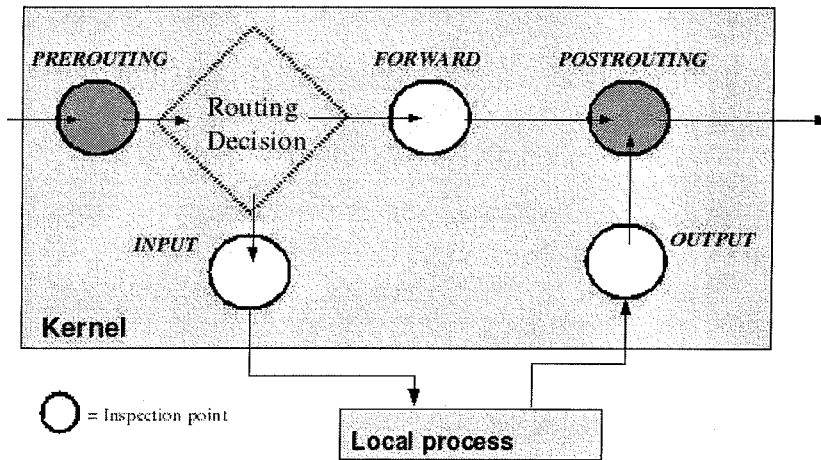
For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A realistic example would be to do something like the above, using a mostly closed approach.

The above example denies access to all TCP-wrapped services for everyone, except those which are explicitly allowed. In this case **ftp** access is allowed to all hosts in the 192.168.0 subnet while **telnet** and **ssh** access are allowed by everyone in the example.com domain as well as host 192.168.2.5. This is a better a method for tightening down a system. It is a simpler, more direct approach and is much easier to maintain.



# Netfilter Packet Flow



12-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**Routing Decision:** If a packet's destination address corresponds to the local system, then packets are routed to be handled there, by a local process. If the packet is to be delivered to another system, and packet forwarding is enabled in the kernel (see `/proc/sys/net/ipv4/ip_forward`) then packets are directed in accordance with the routing table.

Packet filtering takes place within the kernel at the five packet filtering points shown. Note, that the filtering point names are case-sensitive and are all in upper case.

**PREROUTING:** This filtering point deals with packets first upon arrival. (nat)

**FORWARD:** This filtering point handles packets being routed through the local system. (filter)

**INPUT:** This filtering point handles packets destined for the local system, after the routing decision. (filter)

**OUTPUT:** This filtering point handles packets after they have left their sending process, and prior to POSTROUTING. (nat and filter)

**POSTROUTING:** This filtering point handles packets immediately prior to leaving the system. (nat)

## Rule Matching

- Rules in ordered list
- Packets tested against each rule in turn
- On first match, the target is evaluated: usually exits the chain
- Rule may specify multiple criteria for match
- Every criterion in a specification must be met for the rule to match (logical AND)
- Chain policy applies if no match

12-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Rules are in an ordered list. The position of a rule in the list has a bearing on when and if the rule will be used. Packets are tested against each rule (starting at the top of the list) in turn.

Netfilter matching is on a first match basis. If a packet's characteristics match a rule, then the rule's target is evaluated, which usually means that packet checking stops. Packet filtering on some operating systems work on a last-match basis.

If a rule specifies multiple criteria in the match specification, then packets must match every one for the packet to be considered matched by that rule.

If a built-in chain is traversed entirely and no match is found then the chain's default policy applies. In the case of a custom chain, if there is no match then control returns to the chain from which the custom chain was called.

# Rule Targets

- Built-in targets: DROP, ACCEPT
- Extension targets: LOG, REJECT, custom chain
  - REJECT sends a notice returned to sender
  - LOG connects to system log kernel facility
  - LOG match does not exit the chain
- Target is optional, but no more than one per rule and defaults to the chain policy if absent

12-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (819) 754 3700.

Rule targets determine what action to take when a packet matches the rule's selection criteria. Target names qualify the **-j** option of the **iptables** command (think **j** as in *jump*). A target can be a built-in (Base) target, a custom chain or extension target.

The netfilter framework supports DROP and ACCEPT as the two base targets.

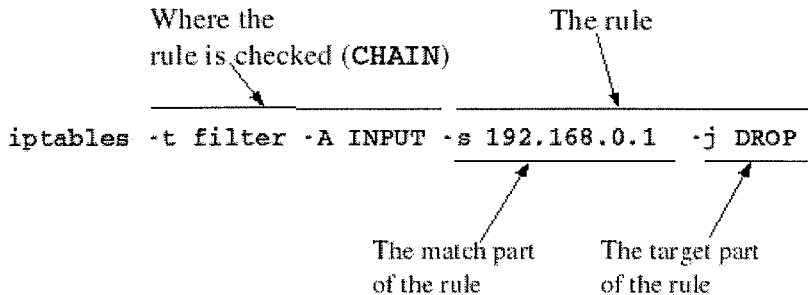
As with packet matching, additional target capability is added through extensions. Unlike packet matching, extension targets are usually implemented in special-purpose kernel modules. For example, the LOG target is implemented through the `ipt_LOG` kernel module. All extension target modules have names starting with `ipt_`.

Most targets do not return; that is, if a packet matches a rule, then checking of that packet ceases (the chain is exited). Some targets, like LOG, do return, after the target's action has been evaluated, control will pass back to the calling chain. There is, for convenience, a RETURN built-in target.

A match in the case of a rule with no target will simply increment a packet and a byte counters associated with that rule. The collection of packet statistics in this manner is probably the only use for rules with no target.

## Simple Example

- An INPUT rule for the filter table:



12-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In this example, the `-A` is for append, indicating that a single rule will be appended (in this case) to the INPUT chain of the filter table. This rule causes any packet with a source address (hence `-s`) of 192.168.0.1 to match and "jump" to its target, DROP, and discarded.

The iptables command provides the user-space door into all the kernel-side capabilities of Netfilter. It is used for all rule administration, and is provided on Red Hat Enterprise Linux as part of the iptables RPM package.

```
[root@stationX]# iptables -t filter -A INPUT -s 192.168.0.1 -j DROP
```

```
[root@stationX]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     port opt source                destination
DROP      all  -- station1.example.com  anywhere
```

The above example shows the **iptables** listing of a rule corresponding to the rule added just before; beware that rules are in effect immediately! Note, it was not necessary to specify the filter table in the command (more on that later). The table could have been made explicit with `-t filter` in the command.

We will learn about the basic syntax elements of the **iptables** command in the following pages. Our first consideration should be whether our system is mostly open (accepting most packets), or mostly closed (denying most packets). This tendency toward open or closed effects not only rules, but most importantly the chain *policy*, in effect when no rule matches or is present. The effect is the *target* for the packet under inspection.

## Basic Chain Operations

- List rules in a chain or table (-L or -vL)
- Append a rule to the chain (-A)
- Insert a rule to the chain (-I)
  - -I CHAIN (inserts as the first rule)
  - -I CHAIN 3 (inserts as rule 3)
- Delete an individual rule (-D)
  - -D CHAIN 3 (deletes rule 3 of the chain)
  - -D CHAIN RULE (deletes rule explicitly)

12-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Several operations may be performed on a chain. Operations effect the chain in the form of rules.

Use -A to append a rule to the end of an existing chain. If the table is not specified, then the *filter* table is assumed.

```
iptables -A INPUT -s 12.34.12.34 -j DROP
```

Insert a rule into a chain as the first, or at a given point with -I..

Use -D to delete rule from a chain based on its sequence number, or explicit specification. Rules are numbered from one.

-F is used to flush, or remove all rules from a chain. This does not reset the chain policy.

```
iptables -t nat -F POSTROUTING
```

To list the contents of the chain (rules and policy), use -L. Using -v with this option displays packet and byte counters, interface(s) and protocols as well.

```
iptables -t filter -L OUTPUT
```

or list the contents of the table (rules and policies).

```
iptables -t filter -L
```

## Additional Chain Operations

- Assign chain policy (`-P CHAIN TARGET`)
  - ACCEPT (default, a built-in target)
  - DROP (a built-in target)
  - REJECT (not permitted, an extension target)
- Flush all rules of a chain (`-F`)
  - Does not flush the policy
- Zero byte and packet counters (`-Z [CHAIN]`)
  - Useful for monitoring chain statistics
- Manage custom chains (`-N, -X`)
  - `-N Your_Chain-Name` (adds chain)
  - `-X Your_Chain-Name` (deletes chain)

12-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Use `-P` to change the default *TARGET*, or policy of a chain. The default installed policy is `ACCEPT`. Only built-in targets (`DROP` and `ACCEPT`) may be a chain policy due to efficiency. While it *is* desirable to inform a client process that its connection is refused (`REJECT`), a new packet must be generated and routed to the client.

Use `-Z` to "zero," or set the byte and packet counters for all rules of a chain to zero (0). This is useful in the systematic collection, and statistical analysis of a rule's effectiveness. Remember that all packets traversing an interface are inspected. A rule which matches no packets requires the same--and important--system resources as a rule that matches frequently. Analysis of rule counters is also helpful as a metric of system and service activity. To display these counters, use the `-v` option to the **iptables** command.

Use `-N` to create a new, empty custom chain. Custom chains do not have a chain policy (see above). Custom chains are useful in the apportioning of rules which more effectively test packets of a specific network (e.g., matching on the interface, the source, or destination address) or service (e.g., matching on a port). Proper implementation of custom chains reduces unnecessary inspection of *all* packets when only a few, well defined groups of packets are required. To remove a custom chain, use `-X`. Built-in chains cannot be *expunged*.

## Common Match Criteria

- IP address or network
  - **-s 192.168.0.0/24**
  - **-d 192.168.0.1**
- Network interface
  - **-i lo**
  - **-o eth1**
- Criteria can be inverted with '!'
  - **-i eth0 -s '!' 192.168.0.0/24**

12-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Most rules in the filter table involve allowing or denying packets based on their source or destination. Below are examples of options that can be used to create such rules.

A packet's source or destination can be specified with **-s** or **-d**, respectively. The option should be followed by an IP address or IP/Netmask combination or hostname. Netmasks can use CIDR (**192.168.0.0/24**) or VLSN (**192.168.0.0/255.255.255.0**) notation. Using a hostname is not recommended because it will just be translated into an IP when the rule is stored anyway.

The following example would allow packets from any address on the 192.168.0.X network through the firewall.

```
# iptables -I INPUT -s 192.168.0.0/24 -j ACCEPT
```

Packets can also be matched based on the physical network interface they are arriving on or leaving through. This is done with the **-i** and **-o** options, respectively. The following command would only allow packets destined for the local network to leave via eth0 (assuming all other packets are denied by default).

```
# iptables -I OUTPUT -o eth0 -d 192.168.0.0/24 -j ACCEPT
```

Another common interface-based rule is the following, which allows all packets arriving on the system's loopback interface through the firewall.

```
# iptables -I INPUT -i lo -j ACCEPT
```

Since only local processes have access to the loopback interface, traffic on lo is usually unfiltered. Thus, many firewall rulesets begin with a rule like the above.

Any match criterion may be negated by prepending **'!'** (with the quotes) to the value. The following example would block all traffic *except* packets from 192.168.0.1:

```
# iptables -I INPUT -s '!' 192.168.0.1 -j DROP
```

## Common Match Criteria continued

- Transport protocol and port
  - `-p tcp --dport 80`
  - `-p udp --sport 53`
  - Port ranges can be specified with *start:end*
- ICMP type
  - `-p icmp --icmp-type host-unreachable`

12-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Most rules in the filter table involve allowing or denying packets based on their source or destination. Below are examples of options that can be used to create such rules.

A packet's source or destination can be specified with `-s` or `-d`, respectively. The option should be followed by an IP address or IP/Netmask combination or hostname. Netmasks can use CIDR (`192.168.0.0/24`) or VLSN (`192.168.0.0/255.255.255.0`) notation. Using a hostname is not recommended because it will just be translated into an IP when the rule is stored anyway.

Packets can also be matched by their source or destination ports. Because port numbers are ambiguous unless associated with a transport protocol (since, e.g., tcp port 53 is distinct from udp port 53) references to ports must always specify a layer-4 protocol with the `-p` option. Destination ports are matched with `--dport` and source ports with `--sport`. Ranges of ports can be listed as "*start\_port:end\_port*". If *end\_port* is left out, it is assumed to be the highest possible port.

The following example would allow tcp packets coming from port 123 of 192.168.0.1 to port 1024 or above of 192.168.0.2:

```
# iptables -I INPUT -p tcp -s 192.168.0.1 --sport 123 -d 192.168.0.2 --dport 1024: -j ACCEPT
```

ICMP packets, which include ping requests and responses, destination-unreachable messages from routers, and many other types of network diagnostic messages, can be selectively filtered by specifying `icmp` as the protocol and using the `--icmp-type` to match specific types. The following examples would explicitly deny ping requests and explicitly allow destination unreachable messages, respectively:

```
# iptables -I INPUT -p icmp --icmp-type echo-request -j DROP
# iptables -I INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

While some networks choose to block pings requests, denying all ICMP packets is not recommended. Certain types, such as destination unreachable messages represent important information that network clients should receive.



## Rules Persistence

- **iptables** is not a daemon, but loads rules into memory and exits
- Rules are not persistent across reboot
  - **service iptables save** will store rules to `/etc/sysconfig/iptables`
  - System V management may be used, and is run before networking is configured
- Conflicts with **ipchains**

12-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux provides a convenient way of saving and restoring the state of all Netfilter rules through the use of a service management script called **iptables**. This allows **iptables** to be operated like a service, even though no "listening" daemon is started.

Backwards compatibility with the earlier **ipchains** facility is provided through a system service of the same name. However, the **ipchains** and **iptables** methods of packet filtering are mutually exclusive. Each facility (**ipchains** and **iptables**) involves the use of its own set of kernel modules. Each service (**ipchains** and **iptables**) depends on the corresponding kernel modules to operate.

## End of Unit 12

- Questions and Answers
- Summary
  - `hosts.allow`, `hosts.deny`
  - `iptables`
  - `service iptables save`

12-17

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 12

## Network Security

---

**Goal:** Deploy basic security infrastructure through `tcp_wrappers` and `iptables`.

**System Setup:** A system with access to the `example.com` network.

Disable the print subsystem before starting this lab:

```
# service cups stop  
# chkconfig cups off
```

This is to cut down on needless network traffic between non-existent printers.

## Sequence 1: Restricting services with tcp\_wrappers

**Scenario:** Remote access services have been installed on systems on your network. As a security minded admin, you want to control this access. Lock the systems down, then provide access as needed.

**Deliverable:** Protected SSH and Telnet services.

**Instructions:**

1. By default, tcpwrappers is "mostly open". Lock down your systems by appending ALL:ALL to your /etc/hosts.deny file. To watch the connect logs to your system open a monitoring window, as follows:

```
xterm -e "cd /var/log; tail -f messages secure" &
```

Attempt to telnet to localhost. This should fail.

2. Look at the log files. There should be an entry for this event. Create a rule to allow the connection.
3. Test ssh to localhost. It should work. Use ssh to connect to address ::1, which is the IPv6 loopback address. Why did it fail? Review the log files and fix the problem.
4. Have a classmate attempt to telnet to your system as an unprivileged account. This should fail. Review the logs and build a rule that will allow only that single machine access.
5. Have a classmate attempt to ssh to your system. The connection should fail. Build a rule that would authorize all systems on the 192.168.0.0/24 network to access your ssh server. Have a classmate attempt to ssh so your system.
6. Next, try to ssh with a link-local IPv6 address. This will also fail. To authorize Scope:Link connections, add [fe80::]/64 to hosts.allow.

## Sequence 2: Applying simple packet filtering to a host

**Scenario:** Your organization has chosen to deploy local firewalls on their servers to protect against threats on the internal network.

**Deliverable:** Packet filter rules successfully limiting connections to your SSH and Telnet servers.

### Instructions:

1. Apply a rule that will **DROP** all traffic to the **INPUT** chain. Attempt to telnet to localhost. This should fail. Just as with `tcp_wrappers`, our first rule needs to grant full access to localhost. Insert a rule above the previous **DROP** rule as follows:

```
# iptables -I INPUT 1 -i lo -j ACCEPT
```

Attempt to telnet to localhost. This should succeed.

2. Execute **host server1**. This should fail. List your **INPUT** rules with the `-v` option. The problem is that the last line is blocking DNS transactions. To see what is being blocked, add a rule to **LOG** packets just above the last **DROP** rule. List your **INPUT** rules again to ensure **LOG** is in the correct place.
3. Execute **host server1**. It should fail, but this time it will generate some errors in the monitoring window. Review the logs, determine the protocol and the source port, and build a rule to allow DNS transactions.
4. Save the **iptables** rules. Open the file in an editor. Notice the last **DROP** rule. Change **DROP** to **REJECT**, save your changes, and restart **iptables**. If they failed to restart, find and fix problem.
5. List the rules with the `-v` option. Did **DROP** change to **REJECT**? In the real world **DROP** is the better choice, but when troubleshooting use **REJECT**.
6. Have a classmate attempt to **SSH** to your system. This should fail. Review the log files and build a rule that will allow all stations on the `192.168.0.0/24` network to connect through your firewall.
7. Try to **SSH** to a classmates system. This should fail. Review the log files and build a rule that will allow your station to **SSH** to any location.
8. Save your current rules. Edit the **iptables** file. After the **DNS** rule (`--sport 53`), add a rule to allow all traffic from `server1.example.com`, and reload **iptables**.

## Sequence 1 Solutions

1. By default, tcpwrappers is "mostly open". Lock down your systems by appending ALL:ALL to your /etc/hosts.deny file.

```
vi /etc/hosts.deny; cat /etc/hosts.deny
... output truncated ...
ALL:ALL
```

To watch the connect logs to your system open a monitoring window, as follows:

```
xterm -e "cd /var/log; tail -f messages secure" &
```

Attempt to telnet to localhost. This should fail.

```
# telnet localhost
... output truncated ...
Connection closed by foreign host.
```

2. Look at the log files. You should see an entry similar to the following:

```
Mar  2 22:09:31 stationX xinetd[3925]: libwrap refused
connection to telnet (libwrap=in.telnetd) from 127.0.0.1
```

This line states that the xinetd process's libwrap object would not launch an in.telnetd connection the address of 127.0.0.1. To authorize this connection, we need an entry in /etc/hosts.allow. Since we know that this is our own localhost IP, it is normal to grant full access to all processes for 127.0.0.1.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
... output truncated ...
ALL:127.0.0.1
```

3. Test ssh to localhost. It should work because ssh is using the same rule as telnet.

```
# ssh localhost
root@localhost's password:
```

Use ssh to connect to address ::1, which is the IPv6 loopback address.

```
# ssh ::1
ssh_exchange_identification: Connection closed by remote host
```

This failed because the original rule is for 127.0.0.1, not ::1. Another rule is needed, or at least another address for the same rule. Remember that IPv6 addresses have to be surrounded by brackets.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
... output truncated ...
ALL:127.0.0.1 [::1]
```

4. Have a classmate attempt to telnet to your system as an unprivileged account. This should fail. You should see a log entry similar to the following:

```
Mar  2 22:59:05 stationX xinetd[4171]: libwrap refused
connection to telnet (libwrap=in.telnetd) from 192.168.0.Y
```

Notice the libwrap= notation in the log entry. This is the name of the process that must be authorized in the hosts.allow file. Add a rule to allow only that single machine to access your system.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
... output truncated ...
ALL:127.0.0.1 [::1]
in.telnetd:192.168.0.Y
```

5. Have a classmate attempt to ssh to you system. The connection should fail. The message, this time, however, is not as obvious:

```
Mar  2 22:38:06 stationX sshd[4126]: refused connect from
::ffff:192.168.0.Y (::ffff:192.168.0.Y)
```

Notice that this message is coming from the sshd process. We could build a rule that would authorize that single system, but instead authorize all hosts on the 192.168.0.0/24 network to access your ssh server.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
... output truncated ...
ALL:127.0.0.1 [::1]
in.telnetd:192.168.0.Y
sshd:192.168.0.
```

(Remember: tcp\_wrappers doesn't use CIDR, so you can't use the /24 notation.)

6. Next, try to ssh with a link-local IPv6 address. This will also fail. To authorize Scope:Link connections, add [fe80::]/64 to hosts.allow.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
... output truncated ...
ALL:127.0.0.1 [::1]
in.telnetd:192.168.0.Y
sshd:192.168.0. [fe80::]/64
```

## Sequence 2 Solutions

1. Apply a rule that will DROP all traffic to the INPUT chain. Attempt to telnet to localhost. This should fail.

```
# iptables -A INPUT -j DROP
# telnet localhost
Trying 127.0.0.1...
```

Just as with tcp\_wrappers, our first rule needs to grant full access to localhost. Insert a rule above the previous DROP rule as follows:

```
# iptables -I INPUT 1 -i lo -j ACCEPT
```

Attempt to telnet to localhost. This should succeed.

2. Execute **host server1**. This should fail.

```
# host station1
;; connection timed out; no servers could be reached
```

List your INPUT rules with the -v option.

```
# iptables -vL INPUT
... output trimmed for brevity ...
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
target prot opt in out source destination
ACCEPT all -- lo any anywhere anywhere
DROP all -- any any anywhere anywhere
```

The problem is that the last line is blocking DNS transactions. To see what is being blocked, add a rule to LOG packets just above the last DROP rule.

```
# iptables -I INPUT 2 -j LOG
```

List your INPUT rules again to ensure LOG is in the correct place.

```
# iptables -vL INPUT
... output trimmed for brevity ...
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
target prot opt in out source destination
ACCEPT all -- lo any anywhere anywhere
LOG all -- any any anywhere anywhere LOG level 4
warning
DROP all -- any any anywhere anywhere
```

3. Execute **host server1**. It should fail, but this time it will generate some errors in the monitoring window. The errors may look similar to the following:

```
Mar 2 23:49:30 stationX kernel: IN=eth0 OUT=
MAC=00:16:3e:03:10:03:00:50:bf:19:7a:07:08:00 SRC=192.168.0.254
```



DST=192.168.0.X LEN=148 TOS=0x00 PREC=0x00 TTL=64 ID=2447 DF ✓  
PROTO=UDP SPT=53 DPT=32770 LEN=128

Notice the fields for PROTO and SPT. These indicate the traffic is UDP entering on port 53.  
To approve these transactions add a rule similar to the following:

```
# iptables -I INPUT 2 -p udp --sport 53 -j ACCEPT
```

(Remember: Order of the rules is important! Make sure this is above both your LOG and DROP rules.)

```
# host server1
server1.example.com has address 192.168.0.254
```

4. Save the iptables rules. Open the file in an editor. Notice the last DROP rule. Change DROP to REJECT, save your changes, and restart iptables.

```
# service iptables save
Saving firewall rules to /etc/sysconfig/iptables: [ OK ]
# cat /etc/sysconfig/iptables | grep "DROP\|REJECT"
-A INPUT -j DROP
# vi /etc/sysconfig/iptables
# cat /etc/sysconfig/iptables | grep "DROP\|REJECT"
-A INPUT -j REJECT
# service iptables restart
Flushing firewall rules: [ OK ]
Setting chains to policy ACCEPT: filter [ OK ]
Unloading iptables modules: [ OK ]
Applying iptables firewall rules: [FAILED]
```

What went wrong? Look at the log window. You may see a message similar to the following:

```
Mar 3 00:09:45 stationX setroubleshoot: SELinux is
preventing /sbin/iptables-restore (iptables_t) "read" to
iptables (etc_runtime_t).
```

In the process of editing the file, the security context has become corrupted. How can we determine the correct security context? We may not have to: Try restorecon on the file.

```
# service iptables start
Applying iptables firewall rules: [ OK ]
```

Whew. That was a close one. Glad we got that fixed.

5. List the rules with the -v option. Did DROP change to REJECT? In the real world DROP is the better choice, but when troubleshooting use REJECT.
6. Have a classmate attempt to SSH to your system. This should fail. Review the log files. There should be an entry similar to the following:

```
Mar 3 00:39:06 stationX kernel: IN=eth0 OUT= ✓
```

```
MAC=00:16:3e:03:10:03:00:0d:60:fa:f5:f2:08:00 SRC=192.168.0.Y ✓  
DST=192.168.0.X LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=11997 DF ✓  
PROTO=TCP SPT=42029 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
```

This time, notice the **PROTO** and **DPT** fields. We could build a set of rules for each machine in the network by using the **SRC** address. Instead build one rule to authorize all the systems on the network. Luckily, iptables supports CIDR, so we could build a rule similar to the following:

```
# iptables -I INPUT 2 -s 192.168.0.0/24 -p tcp --dport 22 -j ✓  
ACCEPT
```

Have a classmate test SSH access to validate the new rule.

7. Try to SSH to a classmates system. This should fail. Another review of the log files will indicate the opposite condition we saw in the previous example. This time, notice the **PROTO** and **SPT** fields. In order to allow your station to connect to any destination we could build a rule similar to the following:

```
# iptables -I INPUT 2 -p tcp --sport 22 -j ACCEPT
```

Connect to a classmate's system to validate the new rule.

8. Save your current rules. Edit the iptables file. After the DNS rule (--sport 53), add a rule to allow all traffic from server1.example.com.

```
# cat /etc/sysconfig/iptables  
... output truncated ...  
-A INPUT -i lo -j ACCEPT  
-A INPUT -p tcp -m tcp --sport 22 -j ACCEPT  
-A INPUT -s 192.168.0.0/255.255.255.0 -p tcp -m tcp --dport 22 ✓  
-j ACCEPT  
-A INPUT -p udp -m udp --sport 53 -j ACCEPT  
-A INPUT -s 192.168.0.254 -j ACCEPT  
-A INPUT -j LOG  
-A INPUT -j REJECT --reject-with icmp-port-unreachable
```

Reload iptables for the new rules to take effect.

# Unit 13

## Network File Sharing Services

13-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[ctraining@redhat.com](mailto:ctraining@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Describe the FTP service
- Describe the NFS Service
- Describe the SMB service

13-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# File Transfer Protocol (FTP)

- The default Red Hat Enterprise Linux FTP server is vsFTPD
- Allows anonymous and user access
- User authentication and transfers are "in the clear"
- Each client connection spawns a new child process
- Server installed with `vsftpd` RPM
- Anonymous download directory is created by the RPM

13-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Very Secure FTP Daemon, **vsftpd**, was designed to be a stable, fast, and scalable FTP daemon. It provides two levels of user access:

- *Anonymous access:* Users can log in as user ftp or as user anonymous to get access to an anonymous ftp site. By default, anonymous users are chrooted into `/var/ftp/` for security.
- *User access:* Users with accounts on the target system can connect via FTP and log in using their username and password. They can download any file they can read, and upload to any directory which they have write access.

# FTP Security

- iptables
  - ftp = tcp:21, ftp-data = tcp:20
  - passive ftp = random ports assigned
  - iptables Connection Tracking
- tcp\_wrappers
  - vsftpd
- SELinux
  - Home directories disabled by default
  - Uploads disabled by default
  - Download directories: public\_content\_t
  - Upload directories: public\_content\_rw\_t

13-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Perhaps the single most frustrating aspect of securing an FTP server is dealing with the firewall configuration. As with NFS, FTP uses a series of ports (20 and 21) for coordinating between the client and server. The transfers are also executed on random ports. Unfortunately, there is not a mechanism to "lock" FTP to specific transfer ports.

In order to navigate an iptables firewall, we use a process called *Connection Tracking*. A kernel module, `ip_conntrack_ftp` is loaded to monitor the data stream to the FTP daemon. Upon seeing a request of PASV, the connection tracking module allows packets to bind to the random port, even though no allow rules exist.

To activate connection tracking, modify `/etc/sysconfig/iptables-config` to reference `ip_conntrack_ftp`. Use **iptables** to allow the initial connection with vsFTPD on port 21, then add another rule to allow `ip_conntrack_ftp` to control the firewall. This is done with the following:

```
# iptables -A INPUT -p tcp --dport 21 -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISH,RELATED -j ACCEPT
```

The vsFTPD daemon uses `tcp_wrappers`:

```
# ldd /usr/sbin/vsftpd | grep libwrap
libwrap.so.0 => /usr/lib64/libwrap.so.0 (0x00002aaaaaf0f000)
# strings /usr/sbin/vsftpd | grep "hosts\..."
```

The default SELinux configuration restricts several of FTP's features. A series of booleans can be used to lessen the access restrictions. Booleans can be activated for the following:

- Allow uploads to `public_content_rw_t` directories

- Allow sharing of NFS directories
- Allow sharing of Samba directories
- Allow users to access home directories

As a worst case scenario, vsFTPD can be run in the less restricted `initrd_t` mode by disabling SELinux's protection of the daemon.

# FTP Configuration

- `/etc/vsftpd/vsftpd.conf`
  - `anonymous_enable=YES`
  - `local_enable=YES`
  - `anon_upload_enable=YES`
  - `ftpd_banner` or `banner_file`

13-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The configuration file for **vsftpd** is `/etc/vsftpd/vsftpd.conf`.

Two access files are used by **vsftpd**. Individual users can be denied access by placing their username in `/etc/vsftpd/ftpusers`. A second file, `/etc/vsftpd/user_list` is only examined if `userlist_enable=YES` is set in `/etc/vsftpd/vsftpd.conf`. It can be used either to list users which will be allowed access or users which will be denied access, depending on whether the option `userlist_deny=NO` is set (default is YES). In order to successfully ftp to the server, users must satisfy the requirements of both access files.

If a file called `.message` exists in a directory, the contents of that file will be displayed to FTP clients accessing that directory.



# Network File Service (NFS)

- The Red Hat Enterprise Linux NFS service is similar to other BSD and UNIX variants
- Stateless file sharing: no *server*
- Shared directories are accessed via **mount**
- NFS is an RPC service and thus requires **portmap**
- Transfers are handled by a set number of **nfsd** processes
- Installed as a base feature

13-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The NFS server software included with Red Hat Enterprise Linux is composed of three facilities, included in the `portmap` and `nfs-utils` rpms:

**portmap:** maps calls made from other machines to the correct RPC service

**nfs** (in kernel): translates NFS requests into requests on the local filesystem

**rpc.mountd:** mounts and unmounts filesystems

These all run as daemons and are started at boot time from the `portmap` and `nfs` System V initialization scripts. Also present will be a set of `nfsd` processes. Each process represents an potential network connection.

```
[root@stationX ~]# grep -i "count=" /etc/init.d/nfs
[ -z "$RPCNFSDCOUNT" ] && RPCNFSDCOUNT=8
[root@stationX ~]# ps -ef | grep nfsd
root      3654      1  0  01:13 ?        00:00:00 [nfsd]
root      3655      1  0  01:13 ?        00:00:00 [nfsd]
root      3658      1  0  01:13 ?        00:00:00 [nfsd]
root      3659      1  0  01:13 ?        00:00:00 [nfsd]
root      3660      1  0  01:13 ?        00:00:00 [nfsd]
root      3661      1  0  01:13 ?        00:00:00 [nfsd]
root      3662      1  0  01:13 ?        00:00:00 [nfsd]
root      3663      1  0  01:13 ?        00:00:00 [nfsd]
```

Red Hat Enterprise Linux supports NFS version 4.0, and uses TCP connections by default

# NFS Security

- **iptables**
  - portmap = tcp:111, nfsd = tcp:2049, udp:2049
  - mountd = random ports assigned by portmap
- **tcp\_wrappers**
  - portmap
- **SELinux**
  - Most file types accessible by default
  - NFS home directories disabled by default
  - public\_content\_t

13-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In order to access NFS shares, a host must establish a connection to **portmap**, which is protected by **tcp\_wrappers**:

```
[root@stationX ~]# ldd /sbin/portmap | grep libwrap
[root@stationX ~]# strings /sbin/portmap | grep "hosts\..*"
/etc/hosts.allow
/etc/hosts.deny
```

This allows for a double layer of security: first **iptables**, second **tcp\_wrappers**. The **mountd** service does not directly use **libwrap.so**, but it is most easily located through **portmap**, which does. Add an entry as below to **/etc/hosts.allow** to allow the 192.168.0.0/24 subnet access to NFS services:

```
portmap: 192.168.0.
```

Note that this does not protect against access by someone who uses **nmap** or a similar tool to determine what ports you are using without reference to the **portmap** service.

Perhaps the single most difficult aspect of securing NFS is the way the **mountd** process negotiates seemingly random port values between the two hosts. To ease **iptables** configuration, Red Hat Enterprise Linux can force **portmap** to allocate specific port addresses.

As of Red Hat Enterprise Linux Version 5, NFS has been restricted by SELinux. Most notably, boolean settings prevent access to **/home**, by default. Another boolean can force NFS into read only mode, regardless of the **/etc/exports** file.

A new context exists for NFS shares, **public\_content\_t**, though most files are available.

## NFS Optional Firewall Ports

- **mountd**, **statd** and **lockd** can be forced to use a static port
- Set the variables in `/etc/sysconfig/nfs`
  - **MOUNTD\_PORT**="655"
  - **STATD\_PORT**="656"
  - **LOCKD\_TCP**PORT="52004"
  - **LOCKD\_UDP**PORT="52004"

13-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To use a firewall with NFS, you must force the services that are normally assigned ports by **portmap** to use static ports. These include the **mountd**, **statd** and **lockd** daemons.

The `nfs` script checks the `/etc/sysconfig/nfs` file for options that it uses when the daemon starts. The **MOUNTD\_PORT**, **STATD\_PORT**, **LOCKD\_TCP**PORT and **LOCKD\_UDP**PORT variables can be used to force these daemons to start on a specific port. The file may not exist on a default installation, so create it and add entries as below.

```
MOUNTD_PORT="655"
STATD_PORT="656"
LOCKD_TCPPORT="52004"
LOCKD_UDPPORT="52004"
```

# NFS Configuration

- /etc/exports

- Share path
- Authorized clients by name or IP

server1.redhat.com or 192.168.0.254

\*.redhat.com or 192.168.0.0/255.255.255.0

- Options must be specified

(ro,sync,root\_squash)

- root mapped to UID 4294967294

13-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Filesystems to be exported via NFS are defined in /etc/exports. Here is an example:

```
/var/ftp/pub          *.example.com(ro,sync)      bigserver.redhat.com(rw,syn
/root/presentations   server2.example.com(rw,sync)
/data                 192.168.10.0/255.255.255.0(sync)
```

Each entry specifies one exported directory and its access permissions. One or more host/permission pairs can be specified, but entries cannot be split into multiple lines. Hostnames can contain wildcards, as in the above example. Wildcards can also be used to match hostnames or domain names: station1\* will match station1, station10, station11, etc.; \*.example.com will match station1.example.com and station1.corp.example.com. The ? wildcard, indicating a match of exactly one character, is also supported.

IP addresses of hosts can be specified individually or using a network/netmask specification. Entries in /etc/exports are exported read-only by default.

Options must not be separated from hostnames with whitespace. If whitespace exists between a hostname and an option, it is treated as two distinct export destinations and the option will apply to a "world" export. This is probably not what is intended.

Entries in /etc/exports are exported with *root\_squashing* turned on. This ensures that requests from the root user on a client machine are denied root access to root-owned files on a server machine. Such requests are mapped instead to a uid such as 65534 or 4294967294. This behavior can be defeated with the *no\_root\_squash* option, but this is not recommended.

**service nfs status** and **exportfs -v** will help confirm proper operation of your NFS server. For more information see the man page for exports.

## NFS Client-side

- Test the connection:
  - **rpcinfo -p *hostname*** connects to **portmap**
  - **showmount -e *hostname*** connects to **nfsd**
- **mount** or **/etc/fstab**
- **autofs** via **/net/*hostname***

13-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**rpcinfo -p *hostname*** is used to probe the portmapper on *hostname* and print a list of all registered RPC services.

**showmount -e *hostname*** is used to display the exports from remote machines (or localhost).

To associate a shared directory on the network with a mount point in your local filesystem, use **mount**. When you mount an exported directory from an NFS server, you can access it as if it was local to your machine. Shares can be mounted manually by **root**, or automatically at boot time.

**/etc/fstab** allows you to specify network directories to be mounted at boot. Here's a sample **fstab** entry that defines a shared filesystem **/var/ftp/pub** on **server1** to be mounted locally as **/mnt/pub**.

```
server1:/var/ftp/pub    /mnt/pub    nfs    defaults    0 0
```

**/etc/init.d/netfs** mounts any network filesystems that are configured to be mounted at boot time, such as the one defined above.

Some NFS-specific options that can be used with **mount** or in **/etc/fstab** include:

- **rsize=8192** and **wsize=8192** - will speed up NFS throughput considerably
- **soft** - processes return with an error on a failed I/O attempt
- **hard** - will block a process that tries to access an unreachable share
- **intr** - allows NFS requests to be interrupted or killed if the server is unreachable
- **noLOCK** - disables file locking (**lockd**) and allows inter-operation with older NFS servers.

The automount facility, **autofs**, provides the ability to mount NFS shares on demand and unmount them when they are idle in a way that is transparent to the end user. Install the **autofs** RPM, then examine **/etc/auto.master** and **/etc/auto.misc** for examples of how **autofs** is configured. **autofs** is a kernel service, but the capability must be enabled by configuring **autofs** to run in the appropriate runlevels.

# Samba (SMB)

- Red Hat Enterprise Linux can act as an SMB client or server
- Shared directories are accessed via **mount**
- The **nmbd** process manages resource browsing and WINS server
- The **smbd** process manages authentication, file transfers, and printer sharing
- Each client connection spawns a new **smbd** process
- Server installed by `samba` RPM

13-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

File and printer sharing is probably the most attractive Samba feature to many users. With this functionality, users can easily retrieve files or print to any printer on the network.

Browsing is the capability of participating systems to view the contents of other participating systems in the same *neighborhood*. With the proper authorization, users may look at and use devices and files of other computers as if they were local.

Name resolution is an integral component of browsing. With name resolution, a computer in the *Network Neighborhood* can receive the neighborhood name of other computers; the *neighborhood* name is not necessarily the network name of the computer. Additionally, name resolution comprises a portion of WINS (Windows® Internet Name Service), which allows centralized mapping of NetBIOS names to IP addresses. This name service is independent of DNS.

When a SMB/CIFS client starts, it may need to know what IP address a specified host is using. The client will broadcast this request on the network and will receive a response from **nmbd**, informing the client of its NetBIOS information.

Broadcast requests can saturate a network, making it virtually unusable. To combat this, Microsoft® developed the WINS protocol, which pulls all the broadcast isolated subnets into a single NetBIOS scope. **nmbd** will act as a WINS server, maintaining a database of all computers on the network. Note that you should not mix the use of the Samba WINS server and the Windows® NT one. In a mixed NT and Samba environment, Samba recommends that you use the NT server's WINS capabilities.

**smbd** provides file space and printer services to clients using the SMB/CIFS protocol. Using the notion of *shares*, or logical volumes of disk or printers, **smbd** can provide access to these facilities with the proper authorization.

# SMB Security

- iptables
  - nmbd = 137/udp, 138/udp
  - smbd = 139/tcp, 445/tcp
- tcp\_wrappers
  - not used
- SELinux
  - Home directories disabled by default
  - public\_content\_t
  - public\_content\_rw\_t

13-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Samba does not use tcp\_wrappers. It does have however a `hosts allow` statement in its global configuration, that provides similar functionality. By default, this option is commented out, meaning Samba operates "wide open".

For normal operations, Samba needs access to four ports on the fire wall. Depending on the application, it may be possible to use fewer. These ports are 137, 138, 39, and 445.

Perhaps the most significant security obstacle to Samba is SELinux. A default Samba installation opens very broad, remote, access to a system. SELinux closes that, by default. A series of booleans can be used to lessen the access restrictions. Booleans can be activated to allow the following:

- Allow sharing of NFS directories
- Allow sharing of user home directories
- Allow writing to `public_content_rw_t` directories
- Allow users to login with CIFS home directories

As a worst case scenario, Samba can be run the less restricted `initrd_t` mode by disabling SELinux's protection of the daemons.

# SMB Configuration

## [global]

- /etc/samba/smb.conf
- [global]
  - workgroup
  - server string
  - hosts allow
  - load printers
  - interfaces

13-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `/etc/samba/smb.conf` file consists of sections and parameters. A section begins with the name of the section in square brackets and continues until the next section begins. Sections contain parameters of the form:

`name = value`

Parameters of an individual section override `[global]` section parameters.

Comments may either be shell or assembly style; that is, all text after a hash (#) or semi-colon (;) to the end of the line are ignored:

`# this is a comment`

`; this is also a comment`

Usually, the comments that begin with a # are informational comments, whereas comments that begin with a ; are code comments: ones that can be used to configure the server.



## SMB Configuration, cont shares

- Shares defined as `[sharename]`
- Predefined shares: `[homes]` and `[printers]`
  - comment - displayed when browsed
  - path - absolute path of share
  - public - share can be accessed by guest
  - browsable - share is visible in browse list
  - writable - resource is read write enabled
  - printable - resource is a printer, not a disk
  - group - files saved with the specified group

13-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The example Samba configuration file shows many different configurations of shares. As a simple example, consider Joe, who wants to share his home directory. Access control would be determined by normal group memberships and file permissions. The system administrator might then add the following to `/etc/samba/smb.conf`:

```
# share Joe's home directory
[joe-home]
    comment = Joe's Home Directory
    path = /home/joe
    public = no
    writable = yes
    printable = no
```

If Joe wanted group-write access permissions for the `devel` group, and read-only access for everyone else, the system administrator might then change the above entry to:

```
# share Joe's home directory
[joe-home]
    comment = Joe's Home Directory
    path = /home/joe
    public = no
    writable = no
    write list = @devel
    printable = no
```

For a more detailed list of options, see **man smb.conf**

# SMB Passwords

- Users must have local accounts
  - Accounts can be translated through `/etc/samba/smbusers`
- Stored in `/etc/samba/smbpasswd`
- Users added with **`smbpasswd -a user`**
- Passwords changed with **`smbpasswd user`**

13-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Most Samba administrators will set `security = user` in their configuration file. Using this setting usually requires the setup of the `smbpasswd` file and possibly the setup of the username map file, `smbusers`.

To add a user, first use the **`smbpasswd`** script:

```
smbpasswd -a joe
```

Remember that `joe` must exist in `/etc/passwd` before **`smbpasswd`** will modify the entry. Use **`smbpasswd user`** for subsequent password changes for all your users.

It is also possible to define user accounts in `/etc/samba/smbpasswd`, but specify that a Primary Domain Controller will manage passwords.

If you choose to use domain or server authentication, also set:

```
# need an appropriate domain name (about four chars)
workgroup = MINE
encrypt passwords = yes
# server(s) to validate this domain; searched in
# order
password server = host1 host2 host3
```

If you choose to use the ads authentication, you must also specify your Kerberos *realm*, and possibly the location of your Active Directory server:

```
realm = YOUR.KERBEROS.REALM
password server = your.kerberos.server
```

With ads, you must also configure your Samba server's account in the Microsoft® ADS domain.

```
[root@stationX]# net ads join -U Administrator
```

If you choose to use share authentication, then public access will be denied to all shares, even if they are explicitly set to allow public access. In this scenario, only shares with specified users or groups will be accessible at all.

# SMB Client-side

- Test the connection:
  - **nmblookup** \\* connects to **nmbd**
  - **smbclient -L** *hostname* connects to **smbd**
  - **smbclient //hostname/share** similar to FTP client
- **mount** or **/etc/fstab**

13-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**nmblookup** queries a WINS server in much the same way that **nslookup** queries a DNS server; the same kind of information -- hostname and IP -- will be returned.

You may specify a user name to **smbclient** with which to connect. If you do not, **smbclient** will use the upper-case version of the **USER** or **LOGNAME** environment variables, in that order, and **PASSWORD** if it exists. If you do specify the **-U** option or the **USER** environment variable is set, then a user name will be formed from all characters up to, but not including, a separating percent (%) symbol. Any characters after the percent symbol will be treated as the password for the user. For example:

```
smbclient -L somehost -U 'bob%foobar'
```

Sometimes you will see *service* used in place of *share*; these two words are synonymous. A path of the form *//machine/service* is called a *UNC* path.

**mount** can use **cifs** or **smbfs** to mount a Samba share. **cifs** is the newer code; **smbfs** is being deprecated. Here is an example:

```
mount -t cifs //stationX /mnt/samba -o user=user,dm=domain,uid=500,file_mode  
44
```

The **mount.cifs(8)** man page give more information on options. **mount.cifs** can be **SUID** root to allow non-root users to mount CIFS shares.

To mount a Samba share automatically at boot time, a line can be added to **/etc/fstab**. This presents a problem where you are required to enter the password for the machine to finish the boot. To alleviate this problem, set the **noauto** mount option.

```
//station1/homes /mnt/homes cifs username=bob,uid=bob,noauto 0 0
```

An alternative is to use a *stash file*:

```
//servername/share /mntpt cifs credentials=/etc/samba/cred.txt 0 0
```

The **/etc/samba/cred.txt** file should only be readable by root. This file is an ASCII text file which contains the following:

username=samba\_username  
password=samba\_password

## End of Unit 13

- Questions and Answers
- Summary
  - /etc/vsftpd/vsftpd.conf
  - /etc/exports
  - rpcinfo, showmount
  - /etc/samba/
  - smbclient

13-17

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 13

## Network File Sharing Services

---

**Goal:** Build skills in deploying and securing NFS, FTP, and SMB services.

**System Setup:** Throughout this lab, the hostnames and domain names that you use will be based on the IP address of your machine. Any time the lab refers to stationX, you should replace X with your station number. Any references to stationY will mean an unprivileged account on a classmate's system, or a remote station identified by your instructor.

Your `tcp_wrappers` configuration should include the following:

```
# cat /etc/hosts.allow
ALL:127.0.0.1 [::1]
sshd:192.168.0. [fe80::]/64
```

Your `iptables` firewall rules should include the following:

```
# cat /etc/sysconfig/iptables
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/255.255.255.0 -p tcp -m tcp
    --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --sport 53 -j ACCEPT
-A INPUT -s 192.168.0.254 -j ACCEPT
-A INPUT -j LOG
-A INPUT -j REJECT --reject-with icmp-port-unreachable
```

## Sequence 1: Implementing FTP Services

**Scenario:** A set of customers need anonymous, read-only, access to a directory of files.

**Deliverable:** A protected FTP server, accessible to remote clients.

### Instructions:

1. Starting with **iptables** active, install the **vsftpd** package. Check the service's status. Ensure it starts at boot time.
2. The RPM will create a `/var/ftp/pub` directory on your system. Copy several files into this directory. Change to `/tmp`, Use ftp to connect to localhost, login as anonymous, and transfer a single file.
3. Use ftp to connect to 192.168.0.X. This should fail. Determine if **vsftpd** uses libwrap. Correct the problem by building a `tcp_wrappers` rule to allow all systems on the 192.168.0.0/24 network to access your server.
4. Now that the service is working locally, connect to a remote station, and attempt to access vsftpd. This should fail. Review the log files and fix the problem.
5. You should now be able to login, however, you can not list files. Furthermore, the error logged in the monitoring window is listing seemingly random ports. Notice the message ftp is displaying:

```
227 Entering Passive Mode (192,168,0,19,112,113)
ftp: connect: Connection refused
```

By entering Passive Mode, the client is trying to bind to random unprivileged ports. This presents a problem in that we can not open the firewall ports, if we don't know the ports the client will request. With NFS, Portmap could force the clients to use the ports we specified. With FTP, we need a different approach.

By adding a Connection Tracking module, we can easily control authorize FTP clients. Edit the `/etc/sysconfig/iptables-config` file. Notice that the first set of comments. Add an entry for `ip_conntrack_ftp` and save the file. For iptables to hand the transactions to the module, we need an additional rule. After the FTP entry (`--dport 21`) in your INPUT chain, add:

```
# iptables -I INPUT 3 -m state --state ESTABLISH,RELATED -j ACCEPT
```

Make sure your rules have been committed to disk, restart iptables (required to load the `ip_conntrack_ftp` module), and try again.

6. Make sure all iptables rules have been saved.

## Sequence 2: Implementing NFS Services

**Scenario:** A group of files need to be shared, read-only, to workstations on your network. The users want seamless access to the data, so an NFS mount is a good solution.

**Deliverable:** A target directory, available to remote clients.

### Instructions:

1. Start with **iptables** active. Launch a monitoring window as follows:

```
# xterm -e "cd /var/log; tail -f messages secure" &
```

2. The components for NFS are installed as base elements of Red Hat Enterprise Linux. What is not implemented by default is port restrictions. Create an `/etc/sysconfig/nfs` file similar to the following:

```
MOUNTD_PORT="655"  
STATD_PORT="656"  
LOCKD_TCP_PORT="4004"  
LOCKD_UDP_PORT="4004"
```

3. By default, nothing on your system is shared via NFS. Make a `/srv/shared` directory and export it to the `192.168.0.0/24` network. Copy some files to the directory for testing purposes.
4. Since we know that NFS requires two services, `portmap` and `nfs`, ensure that both services are running and configured to launch at boot time.
5. Use **`rpcinfo -p localhost`** to verify that you can access `portmap`. Notice that it reports `mountd` locked to the port you specified. Execute **`showmount -e localhost`** to verify you can access `nfsd`.
6. At this point the server is working locally. Use SSH to connect to a remote station and attempt **`/usr/sbin/rpcinfo -p 192.168.0.X`**. This should fail. Review the error logs, fix the problem, and try again. Most likely you will have to fix two problems to get the command to work.
7. As you can see we had to open the firewall for port 111 to support `portmap`. We will need more rules to support the other NFS daemons. Add a UDP rule for `portmap`. Add a rule for the `nfsd`'s, which are using `TCP:2049`. Since we locked to ports 655, 656, and 4004 in our first step, we will need UDP rules for each. (Remember: there is a way of adding a range of ports to a rule.) What about the other 4004? What do we need for it?

Once the rules are built, attempt **`/usr/sbin/showmount -e 192.168.0.X`** from station Y.

8. To access an NFS share, we need to mount the server. Luckily, `autofs` will let us browse NFS shares. Execute **`ls /net/stationX/shared`**. How would you rate the response time? Look at the log files. What might speed up the connection?
9. Make sure all **`iptables`** rules have been saved.



## Sequence 3: Implementing SMB Services

**Scenario:** Users need access to a target directory from their "alternative platform" desktops. Share a directory for guest access.

**Deliverable:** A target directory, readable by a Samba guest, and mountable by an authenticated user.

### Instructions:

1. Starting with **iptables** active, install the samba package. Check the service's status. Ensure it starts at boot time.
2. By default, Samba will attempt to share user home directories, but will be blocked by SELinux. Make a directory `/srv/rh300`, to be shared. Open the `smb.conf` file, set the workgroup name to `RH300` and create a share named `rh300` with a path of `/srv/rh300`. Make the share accessible to guest users and writable to real users. Reload the service
3. Use **smbclient -NL** to query to localhost. Since Samba does not implement libwrap, we do not need to make any changes to `hosts.allow`. Try **smbclient -NL 192.168.0.X**. This should also work.
4. Now that the service is functioning locally, connect to a remote station, and execute **smbclient -NL 192.168.0.X**. This should fail. Review the log files. As with earlier exercises, you will have to add a firewall rule to **iptables**. In fact, you will rules for a total of four ports. Be proactive: construct all the rules before trying again.

Want a challenge? Can you open all four ports with two rules?

5. Once the share is visible, use **smbclient //stationX/rh300** to access the share. You will be prompted for a password: just press *Enter*. At the `smb: \>` prompt, attempt a listing. This should fail. Examine the log window. You should see an entry similar to the following:

```
Mar 21 12:39:18 localhost setroubleshoot: SELinux is preventing ✓  
samba  
(/usr/sbin/smbd) "read" to rh300 (var_t). For complete SELinux ✓  
messages, run  
sealert -l de3b262e-5355-4436-ab5c-ddc0a0f8374a
```

This is an `setroubleshootd` message. Notice the last portion of the message starting at "sealert". Run this command in another terminal window. Look at the *Allowing Access* section of the report. It is identifying the correct security context for the share. Execute the following:

```
# chcon -R -t samba_share_t /srv/rh300
```

6. Again, use **smbclient** to access the share. Attempt to list the directory. This should succeed. Execute the following:

```
smb: \> lcd /etc  
smb: \> put passwd
```

This should fail. Look at the permissions on `/srv/rh300`. You should notice that the directory is not writable by guest. Temporarily, grant other's write permission, and try the transfer again. This should succeed. Restore the original permissions.

7. To access an NFS share, we needed to mount the server. With NFS, we were able to use autofs. Unfortunately, a quick listing of `/net/192.168.0.X/srv` does not show our `rh300` share. If we had root privileges on the remote system we could mount it directly. Ask your classmate for their root password. Hopefully, they won't give it to you. The good news is that student should have sudo permission to mount, as a result of an earlier lab. Access `stationY` as student, create a mount point of `~/stationX`. attempt to access your share as follows:

```
$ sudo mount //stationX/rh300 ~/stationX
```

You will be prompted for two passwords, the first is the user's sudo password, the second is the Samba password (just press *Enter* for guest access.) This should fail. Guest accounts can not mount Samba shares.

8. Assign student the Samba password of "samba" and attempt to mount as follows:

```
$ sudo mount //stationX/rh300 ~/stationX -o username=student
```

Provide the correct password. This should succeed. Copy the `/etc/group` file to the share. This should fail. Examine the permissions on the directory. You will notice student does not have write privileges. Change the directory's ownership, and try again.

## Sequence 1 Solutions

1. Starting with **iptables** active, install the **vsftpd** package.

```
# service iptables start
Applying iptables firewall rules: [ OK ]
# yum install -y vsftpd
... output truncated ...
Installed: vsftpd.x86_64 0:2.0.5-9
Complete!
```

Check the service's status. Ensure it starts at boot time.

```
# service vsftpd status
vsftpd is stopped
# service vsftpd start
Starting vsftpd for vsftpd: [ OK ]
# chkconfig vsftpd --list
vsftpd    0:off  1:off  2:off  3:off  4:off  5:off  6:off
# chkconfig vsftpd on
```

2. The RPM will create a `/var/ftp/pub` directory on your system. Copy several files into this directory. Change to `/tmp`, Use `ftp` to connect to localhost, login as anonymous, and transfer a single file.

```
# cp /etc/hosts* /var/ftp/pub
# ls /var/ftp/pub
hosts hosts.allow hosts.deny
# cd /tmp
# ftp localhost
... output truncated ...
ftp> get hosts
local: hosts remote: hosts
227 Entering Passive Mode (127,0,0,1,97,242)
150 Opening BINARY mode data connection for hosts (252 bytes).
226 File send OK.
252 bytes received in 6.1e-05 seconds (4e+03 Kbytes/s)
ftp> quit
```

3. Use `ftp` to connect to 192.168.0.X. This should fail. Determine if **vsftpd** uses `libwrap`.

```
# ftp 192.168.0.19
Connected to 192.168.0.19.
421 Service not available.
ftp> quit
# ldd /usr/sbin/vsftpd | grep libwrap
libwrap.so.0 => /usr/lib64/libwrap.so.0 (0x00002aaaaaf0f000)
# strings /usr/sbin/vsftpd | grep "hosts\..."
```

Correct the problem by building a `tcp_wrappers` rule to allow all systems on the 192.168.0.0/24 network to access your server.

```
# vi /etc/hosts.allow; grep vsftpd /etc/hosts.allow
vsftd : 192.168.0.
# ftp 192.168.0.19
... output truncated ...
Name (192.168.0.19:root):
```

4. Now that the service is working locally, connect to a remote station, and attempt to access vsftpd. This should fail.

```
[stationY]$ ftp 192.168.0.19
ftp: connect: Connection refused
```

In the monitoring window, there should be an log entry similar to the following:

```
Mar  3 12:10:56 stationX kernel: IN=eth0 OUT=
MAC=00:16:3e:03:10:03:00:0d:60:fa:f5:f2:08:00 SRC=192.168.69.32
DST=192.168.0.19 LEN=52 TOS=0x10 PREC=0x00 TTL=64 ID=20040 DF
PROTO=TCP SPT=52631 DPT=21 WINDOW=365 RES=0x00 ACK FIN URGP=0
```

Notice the fields for PROTO and DPT. From this information we could build a rule similar to the following:

```
# iptables -I INPUT 2 -p tcp --dport 21 -j ACCEPT
```

With the rule in place, try to connect:

```
[stationY]$ ftp 192.168.0.19
... output truncated ...
Name (192.168.0.19:student): anonymous
```

5. You should now be able to login, however, you can not list files. Furthermore, the error logged in the monitoring window is listing seemingly random ports. Notice the message ftp is displaying:

```
227 Entering Passive Mode (192,168,0,19,112,113)
ftp: connect: Connection refused
```

By entering Passive Mode, the client is trying to bind to random unprivileged ports. This presents a problem in that we can not open the firewall ports, if we don't know the ports the client will request. With NFS, Portmap could force the clients to use the ports we specified. With FTP, we need a different approach.

By adding a Connection Tracking module, we can easily control authorize FTP clients. Edit the /etc/sysconfig/iptables-config file. Notice that the first set of comments. Add an entry for ip\_conntrack\_ftp and save the file.

```
# vi iptables-config
# grep track iptables-config
IPTABLES_MODULES="ip_conntrack_netbios_ns ip_conntrack_ftp"
```

For iptables to hand the transactions to the module, we need an additional rule. After the FTP entry (--dport 21) in your INPUT chain, add:

```
# iptables -I INPUT 3 -m state --state ESTABLISH,RELATED -j  
ACCEPT
```

Make sure your rules have been committed to disk, restart iptables (required to load the ip\_conntrack\_ftp module), and try again.

```
# service iptables save  
# service iptables restart  
[stationY]$ ftp 192.168.0.X  
... output truncated ...  
ftp> ls  
227 Entering Passive Mode (192,168,0,X,28,201)  
150 Here comes the directory listing.  
drwxr-xr-x    2 0      0      4096 Mar 03 16:42 pub
```

6. Make sure all iptables rules have been saved.

## Sequence 2 Solutions

1. Start with **iptables** active. Launch a monitoring window as follows:

```
# xterm -e "cd /var/log; tail -f messages secure" &
```

2. The components for NFS are installed as base elements of Red Hat Enterprise Linux. What is not implemented by default is port restrictions. Create an `/etc/sysconfig/nfs` file similar to the following:

```
MOUNTD_PORT="655"
STATD_PORT="656"
LOCKD_TCPPORT="4004"
LOCKD_UDPPORT="4004"
```

3. By default, nothing on your system is shared via NFS. Make a `/srv/shared` directory and export it to the 192.168.0.0/24 network. Copy some files to the directory for testing purposes.

```
# mkdir /srv/shared
# vi /etc/exports; cat /etc/exports
/srv/shared 192.168.0.0/255.255.255.0(ro)
# cp /etc/hos* /srv/shared
```

4. Since we know that NFS requires two services, portmap and nfs, ensure that both services are running and configured to launch at boot time.

```
# service portmap status
portmap (pid 9389) is running...
# chkconfig portmap --list
portmap 0:off 1:off 2:off 3:on 4:on 5:on 6:off
# service nfs status
... output truncated ...
nfsd is stopped
# service nfs start
Starting NFS services: [ OK ]
... output truncated ...
# chkconfig nfs --list
nfs 0:off 1:off 2:off 3:off 4:off 5:off 6:off
# chkconfig nfs on
```

5. Use **rpcinfo -p localhost** to verify that you can access portmap.

```
# rpcinfo -p localhost
... output truncated ...
100005 1 udp 655 mountd
```

Notice that it reports mountd locked to the port you specified. Execute **showmount -e localhost** to verify you can access nfsd.

```
# showmount -e localhost
Export list for localhost:
```

/srv/shared 192.168.0.0/255.255.0.0

6. At this point the server is working locally. Use SSH to connect to a remote station and attempt `/usr/sbin/rpcinfo -p 192.168.0.X`. This should fail.

```
[stationY]$ /usr/sbin/rpcinfo -p 192.168.0.19
rpcinfo: can't contact portmapper: RPC: Remote system error -
Connection refused
```

Review the error logs. There should be an entry similar to the following:

```
Mar  3 14:25:52 stationX kernel: IN=eth0 OUT=
MAC=00:16:3e:03:10:03:0d:60:fa:f5:f2:08:00 SRC=192.168.0.Y
DST=192.168.0.X LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=11349 DF
PROTO=TCP SPT=57053 DPT=111 WINDOW=5840 RES=0x00 SYN URGP=0
```

Notice the **PROTO** and **DPT** fields. This information could be used to construct a rule similar to the following:

```
# iptables -I INPUT 2 -p tcp --dport 111 -j ACCEPT
```

Attempt to connect.

```
[stationY]$ /usr/sbin/rpcinfo -p 192.168.0.19
No remote programs registered.
```

Notice that it is a different message this time. Last time the connection was refused. This time you are reaching portmap, but it is refusing to give you any information. Look at the logs for a message similar to the following:

```
Mar  3 14:34:00 stationX portmap[9931]: connect from 192.168.0.Y
to dump(): request from unauthorized host
```

Add an entry for portmap in `hosts.allow` for the local network, and try the command again.

```
# vi /etc/hosts.allow; cat /etc/hosts.allow
portmap:192.168.0.
[stationY]$ /usr/sbin/rpcinfo -p 192.168.0.19
... output truncated ...
100005      3      tcp      655      mountd
```

7. As you can see we had to open the firewall for port 111 to support portmap. We will need more rules to support the other NFS daemons. Add a UDP rule for portmap.

```
# iptables -I INPUT 2 -p udp --dport 111 -j ACCEPT
```

We could try the `showmount` command, watch it fail, and continue to build rules until we were able to get it to work, or we can think ahead as to what we'll need. We know that we will also need a TCP rule for 2049. Since we locked ports 655, 656, and 4004 in our first step, we will need a UDP rule for each. (Remember: there is a way of specifying a range of ports in a

rule.) Create another rule to cover UDP for the three ports. We will need rules similar to the following:

```
# iptables -I INPUT 2 -p tcp --dport 2049 -j ACCEPT
# iptables -I INPUT 2 -p udp --dport 655:656 -j ACCEPT
# iptables -I INPUT 2 -p udp --dport 4004 -j ACCEPT
# iptables -I INPUT 2 -p tcp --dport 655:656 -j ACCEPT
# iptables -I INPUT 2 -p tcp --dport 4004 -j ACCEPT
```

Save the rules and attempt **/usr/sbin/showmount -e 192.168.0.X** from station Y.

```
$ /usr/sbin/showmount -e 192.168.0.19
Export list for 192.168.0.19:
/srv/shared 192.168.0.0/255.255.255.0
```

8. Time to retrieve a file. To access an NFS share, we need to mount the server. Luckily, autofs will let us browse NFS shares. Execute **ls /net/stationX/shared**.

```
# ls /net/192.168.0.19/srv/shared
hosts.allow hosts.deny
```

It may take a minute or so, but you should get a directory listing. Once connected, you should have rapid responses, but the initial connection maybe slow. Look at the log files. You may see another error for UDP 2049. Build an additional rule, and see if connection time is improved.

```
# iptables -I INPUT 2 -p udp --dport 2049 -j ACCEPT
```

9. Make sure all **iptables** rules have been saved.



## Sequence 3 Solutions

1. Starting with **iptables** active, install the samba package. Check the service's status. Ensure it starts at boot time.

```
# service iptables start
Setting chains to policy ACCEPT: filter          [ OK ]
# yum install -y samba
... output truncated ...
Installed: samba 0:3.0.23c-2
Complete!
# service smb status
smbd is stopped
nmbd is stopped
# service smb start
Starting SMB services:                          [ OK ]
Starting NMB services:                          [ OK ]
# chkconfig smb --list
smb          0:off  1:off  2:off  3:off  4:off  5:off  6:off
# chkconfig smb on
```

2. By default, Samba will attempt to share user home directories, but will be blocked by SELinux. Make a directory `/srv/rh300`, to be shared. Open the `smb.conf` file, set the workgroup name to `RH300` and create a share named `rh300` with a path of `/srv/rh300`. Make the share accessible guest users and writable to real users.

```
# mkdir /srv/rh300
# vi /etc/samba/smb.conf
# grep "RH300" /etc/samba/smb.conf
    workgroup = RH300
# tail -5 /etc/samba/smb.conf
[rh300]
    path=/srv/rh300
    public=yes
    writable=yes
    browsable=yes
```

Reload the service

```
# service smb reload
Reloading smb.conf file:                        [ OK ]
```

3. Use **smbclient -NL** to query to localhost. Since Samba does not implement libwrap, we do not need to make any changes to `hosts.allow`. Try **smbclient -NL 192.168.0.X**. This should also work.

```
# smbclient -NL localhost
... output truncated ...
        rh300                Disk
... output truncated ...
```

4. Now that the service is functioning locally, connect to a remote station, and execute **smbclient -NL 192.168.0.X**. This should fail.

```
[stationY]$ smbclient -NL 192.168.0.19
Error connecting to 192.168.0.19 (Connection refused)
Connection to 192.168.0.19 failed
```

Review the log files. As with earlier exercises, you will have to add a firewall rule to **iptables**. In fact, you will need rules for a total of four ports. Be proactive: construct all the rules before trying again.

Want a challenge? Can you open all four ports with two rules?

```
iptables -I INPUT 2 -p tcp --dport 445 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 137 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 138 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 139 -j ACCEPT
```

As an alternative, you could have made combined ports 137, 138, and 139 as follows:

```
iptables -I INPUT 2 -p tcp --dport 137:139 -j ACCEPT
```

With the rules in place, try again from stationY.

```
$ smbclient -NL localhost
... output truncated ...
      rh300          Disk
... output truncated ...
```

5. Once the share is visible, use **smbclient //stationX/rh300** to access the share. You will be prompted for a password: just press *Enter*. At the smb: \> prompt, attempt a listing. This should fail.

```
# smbclient //localhost/rh300
Password:
Anonymous login successful
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.23c-2]
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
```

Examine the log window. You should see an entry similar to the following:

```
Mar 21 12:39:18 localhost setroubleshoot: SELinux is preventing
samba
(/usr/sbin/smbd) "read" to rh300 (var_t). For complete SELinux
messages, run
sealert -l de3b262e-5355-4436-ab5c-ddc0a0f8374a
```

This is an setroubleshoot message. Notice the last portion of the message starting at "sealert". Run this command in another terminal window.

```
# sealert -l de3b262e-5355-4436-ab5c-ddc0a0f8374a
```

Look at the *Allowing Access* section of the report It is identifying the correct security context for the share. Execute the following:

```
# chcon -R -t samba_share_t /srv/rh300
```

6. Again, use **smbclient** to access the share. Attempt to list the directory. This should succeed. Execute the following:

```
smb: \> lcd /etc
smb: \> put passwd
NT_STATUS_ACCESS_DENIED opening remote file \passwd
```

This should fail. Look at the permissions on `/srv/rh300`. You should notice that the directory is not writable by guest. Temporarily, grant other's write permission, and try the transfer again. This should succeed. Restore the original permissions.

```
# chmod o+w /srv/rh300
```

```
smb: \> put passwd
putting file passwd as \passwd (753.9 kb/s) (average 1853.2
kb/s)
```

```
# chmod o-w /srv/rh300
```

7. To access an NFS share, we needed to mount the server. With NFS, we were able to use autofs. Unfortunately, a quick listing of `/net/192.168.0.X/srv` does not show our `rh300` share. If we had root privileges on the remote system we could mount it directly. Ask your classmate for their root password. Hopefully, they won't give it to you. The good news is that student should have sudo permission to mount, as a result of an earlier lab. Access `stationY` as student, create a mount point of `~/stationX`. attempt to access your share as follows:

```
$ sudo mount //stationX/rh300 ~/stationX
```

You will be prompted for two passwords, the first is the user's sudo password, the second is the Samba password (just press *Enter* for guest access.)

```
$ sudo mount //stationX/rh300 ~/stationX
Password: password
Password:
mount error 13 = Permission denied
Refer to the mount.cifs(8) manual page (e.g.man mount.cifs)
```

This should fail. Guest accounts can not mount Samba shares.

8. Assign student the Samba password of "samba".

```
# smbpasswd -a student
```

New SMB password: **samba**  
Retype new SMB password: **samba**  
Added user student.

Attempt to mount as follows:

```
$ sudo mount //stationX/rh300 ~/stationX -o username=student
```

Provide the correct passwords. This should succeed. Copy the /etc/group file to the share.  
This should fail.

```
$ cp /etc/group ~/stationX  
cp: cannot create regular file: permission denied
```

Examine the permissions on the directory. You will notice student does not have write privileges. Change the directory's ownership, and try again.

```
# chown student /srv/share
```

```
$ cp /etc/group ~/stationX
```

# Unit 14

## Network Infrastructure

14-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[ctraining@redhat.com](mailto:ctraining@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Configure network system logging
- Set up an Installation server
- Set up a Yum server
- Set up Dynamic Host Configuration Protocol
- Configure Network Time Protocol
- Configure the Domain Name Service
- Secure infrastructure services

14-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Enabling Network Logging

- By default, syslogd only listens to localhost
- `/etc/sysconfig/syslog`
- Add `-r` to `SYSLOGD_OPTIONS`
- On the clients, add `@10.11.12.13` as a destination
- Best practice to log both locally and remotely

14-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Network Installation Server

- Necessary for network-based installs
- Often faster than CDROM-based installation methods
- Provides an easy distribution platform for the enterprise
- Shares the RedHat directory via NFS, FTP and/or HTTP
- Can be used as a yum repository

14-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

A network installation server can be easily created from a host running an NFS, FTP, and/or HTTP server.

To create an installation server, select one or more installation protocols:

If using anonymous FTP, copy each install CD's contents to the publicly-accessible FTP directory. It is safe to overwrite files such as TRANS.TBL.

**cp -a /mnt/cdrom/. /var/ftp/pub**

If using HTTP, either use the above technique to copy the contents to a location in your web file tree. If you choose to provide installation services via both FTP and HTTP, create a symbolic link from your web file tree to your existing FTP directory:

```
[root@stationX ~]# ln -s ../../ftp/pub /var/www/html/pub
```

```
[root@stationX ~]# chcon -h -R -t httpd_sys_content_t /var/ftp/pub
```

If using NFS, create an entry in /etc/exports to share the pub directory, for example:

```
/var/ftp/pub hosts.and.or.networks(ro)
```

Restart the servers you have chosen to configure, if necessary.



## Creating a private repository

- Create a directory to hold your packages
- Make this directory available by http/ftp
- Install the **createrepo** RPM
- Run **createrepo -v /package-directory**
- This will create a **repodata** subdirectory and the needed support files
- To support Anaconda on the same server:
  - `cp /package-directory/repodata/comps*.xml /tmp`
  - **createrepo -g /tmp/comps\*.xml /package-directory**

14-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The **createrepo** creates the support files necessary for a yum repository. These files will put into the **repodata** subdirectory.

**repomd.xml** - The **repomd.xml** contains times tamps and checksum values for the other three files. Once a client has established a connection with a server, it caches all files, and only refreshes the cache if **repomd.xml** indicates the repository has changed.

**primary.xml.gz** - The **primary.xml.gz** file contains the list of all the RPMs in the repository, as well as dependency information. It also contains the information that would normally be returned by **rpm -qlp**.

**filelists.xml.gz** - The **filelists.xml.gz** file contains a list of all the files in all the RPMS. This is used by queries such as **yum whatprovides**.

**other.xml.gz** - The **other.xml.gz** file contains additional information, including the change logs for the RPMs.

**comps.xml** - The optional **comps.xml** file contains information about package groups. This allows group installations and optimizes dependency resolution.

The addition or deletion of files within the repository requires a **createrepo** to be run again.

# Configuring an IPv4 DHCP Server

- Configure the server in `/etc/dhcpd.conf`
- Sample configuration provided in `/usr/share/doc/dhcp-version/dhcpd.conf.sample`
- There must be at least one subnet block, and it must correspond with configured interfaces.
- Run **service dhcpd configtest** to check syntax

14-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

To configure a **dhcpd** server, first use **ifconfig** to verify that a BROADCAST address is specified in your network configuration; Initial DHCP requests in IPv4 are broadcast and not sent to a specific server.

Next, edit the `/etc/dhcpd.conf` file. You'll need to configure lease times, optional subnet masks, router addresses, DNS servers, as well as IP addresses or ranges of addresses for your clients. Leased IP addresses are kept in `/var/lib/dhcpd/dhcpd.leases` as they are assigned. Refer to the sample configuration file under `/usr/share/doc/dhcp-version/`, or the manual page (**man 5 dhcpd.conf**) when creating your site-specific DHCP server. Here is a sample:

```
# global definitions
ddns-update-style none;          # turn off DDNS updates; required option
option domain-name "example.com"; # domain name given to client
option domain-name-servers 192.168.0.254;
default-lease-time 21600;        # seconds till expire
max-lease-time 43200;            # maximum lease time
subnet 192.168.0.0 netmask 255.255.255.0
{
# definitions in this block applicable only to given net
  option routers 192.168.0.253;    # local gateway
  option subnet-mask 255.255.255.0; # local subnet mask

  range 192.168.0.2 192.168.0.250; # Range configuration DHCP

  host station1 # static configuration for each host BOOTP
  {
    hardware ethernet 00:a0:cc:3d:0b:39;
    fixed-address 192.168.0.1;
  }
}
```

`/etc/sysconfig/dhcpd` can be used to configure **dhcpd** by setting the **DHCPDARGS** variable. The following would only run the **dhcpd** server on `eth0`:

```
DHCPDARGS="eth0"
```

*Best Practice:* Always run **service dhcpd configtest** after editing `/etc/dhcpd.conf` since configuration errors can prevent dhcpd from starting.

# Basic Design of NTP

- Any NTP client is a potential server
  - "stratum-1" has radio clock or atomic clock
  - "stratum-2" server
    - Is a client of a stratum-1 server
    - May serve time data to stratum-2 or higher (stratum-3, etc.) clients/servers
- Work group usually has 3 low-stratum servers most clients use for time

14-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The lower the stratum of a NTP server, the closer it is to a highly accurate source of time. A stratum-1 server has direct access to a good time source, such as an atomic clock or a WWV, or GPS radio receiver. The highest stratum allowed is stratum-15; a "stratum-16" server is not synchronized. It is rare for an organization to use more than two or three strata internally.

NTP clients will send a synchronization request to their server(s) every fifteen minutes or so – more frequently if the service has started recently or there have been communication problems with the server(s). The clients will attempt to determine the most accurate response based on a number of criteria and use that response to synchronize their system clocks.

An organization usually arranges to have three low-stratum servers as primary time sources. The majority of client workstations will then be told to synchronize from those three servers. So if an organization has three central time servers with radio clocks at stratum-1, then the rest of the client workstations synchronizing from them are at stratum-2. The central servers may not have radio clocks, but may themselves synchronize from a public stratum-1 or stratum-2 time server. For example, the central servers might synchronize from public stratum-2 servers (placing themselves at stratum-3) and the organization's client workstations at stratum-4.

Having three central time servers allows clients to reject bogus synchronization messages if one of the servers' NTP daemons or clocks malfunctions. It is possible for a client to synchronize with fewer time servers if necessary but it is less secure.

# Server Configuration

- Similar to client with three time sources
- Peer with the other two work group servers
- Third: an external *server*
- Radio clock or GPS (uses clock driver)
- Public NTP server
- Last resort: local system clock (LCL)
- Inaccurate source of time; use fudge to advertise at a very high stratum (stratum-10 or higher)

14-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Basic configuration of an organizational server is very similar to that of a end-user client. Again, if possible, you want to have at least three NTP servers as time sources. Two of these are normally the other two organizational servers; these are configured as *peers*, which allows them to synchronize with each other or provide time data to each other. This is done by using the keyword *peer* instead of the keyword *server*.

The third server should be an external time source and should be different for each time server. These are identified with the *server* keyword and can either be public NTP servers with access to a good time source or device drivers to access a hardware clock attached to the local server. These drivers can usually be distinguished as they are given a fake IP address on the 127/8 subnet reserved for the localhost.

LCL is a special driver for the local hardware clock. If there is a system that cannot get time from the Internet and an affordable and accurate clock is not available, the server could use the motherboard hardware clock as an inaccurate time source. Since motherboard clocks tend to be very low quality and gain or lose time quickly, if one is set up, use the *fudge* command so that the server advertises it at a very high (and therefore unreliable) stratum, at least 10.

```
peer 192.168.0.2          # a server peer
server 10.15.0.4          # a public NTP server on 10.15.0.4
server 127.127.29.0       # a Trimble GPS (on a serial port)
server 127.127.1.0        # the motherboard hardware clock
fudge 127.127.1.0 stratum 10 # ...which is advertised as stratum-10
```

# DNS Overview

- Resolves hostnames into IP addresses (forward lookup)
- Resolves IP addresses into hostnames (reverse lookup)
- Allows machines to be logically grouped
  - *hostname.domain.tld*
  - *hostname.subdomain.domain.tld*
- Each name server is responsible for a portion of the names pace, called a *zone*
- DNS queries *forward* up the tree
- Name servers *cache* the responses

14-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The primary purpose of the Domain Name Service (DNS) is to translate hard to remember IP addresses into easier to remember names. In some cases it is helpful to reverse the process, so DNS also has the ability to translate IP addresses back to names. This was originally done by local files, such as the `/etc/hosts`, but over time the files grew too large to be easily maintained. Today's DNS allows for queries of a dispersed, cooperative database network, to ensure that the names and numbers are reasonably synchronized.

Domain names are separated by dots, with the topmost element on the right, whereas IP addresses have the topmost element on the left. Each element may be up to 63 characters long; the entire name may be at most 255 characters long. Letters, numbers, or dashes may be used in an element.

At the top of the database tree are the *root* servers. These servers are responsible for storing information about the *top level domain* servers. The TLD servers (.com, .net, .org, etc) are responsible for storing information about the domains in their branch of the DNS tree. Each domain, in turn, is managed by one or more *name servers*. The name servers are responsible for one or more *zones*, which would contain at least one hostname, each.

It is possible that a zone may contain a reference to another name server, further down the tree. These name servers, would manage a zone, which would be a *subdomain* or the parent server. Any host in that name server's subdomain would also be said to be in that name server's zone.

Host names map to IP addresses in a many-to-many relationship. A host name may have one or more IP addresses. Conversely, a particular IP address may have multiple host names associated with it.

Since the database is distributed across thousands of machines, no one server has a full record of all names and addresses. As such, when a client wants to execute a DNS query, it *forwards* the request to a name server. The name servers in turn forward requests up the tree, toward the root servers. When a server is founded that has a response to the query, the response is sent back to the name server, which *caches* the response. Following request would be served from the cache. To ensure the data does not become stale,

name servers can only cache responses for a certain amount of time. This is called the *time-to-live* value, or TTL.

# Berkeley Internet Name Domain

- Often called *BIND*, runs as **named**
- Provides forward lookup, reverse lookup, forwarding, and caching
- Runs in a chrooted environment
  - `/var/named/chroot`
- Global configuration:
  - `/var/named/chroot/etc/named.conf`
- Can act as both a master and a slave server concurrently, for different domains
  - `/var/named/chroot/var/named/*.zone`
  - `/var/named/chroot/var/named/slaves/*.zone`

14-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Red Hat Enterprise Linux uses BIND 9 for DNS. The BIND 9 service runs as the **named** daemon and provides both forward and reverse lookup services. It can run as a caching only name server, meaning it is not responsible for a zone of hostnames. This only requires the installation of the caching-nameserver RPM and virtually no configuration.

In order to manage a zone, install the `bind` and `bind-chroot` RPMs. The `bind-chroot` package will modify the `/etc/sysconfig/named` file to include the `ROOTDIR` variable:  
`#ROOTDIR=/var/named/chroot`

When this variable is set the `named` process id chrooted to the directory specified in the variable prior to reading any configuration files. The default chroot directory is `/var/named/chroot`. This means that all of BIND's configuration files will be stored relative to this directory. For example, the main configuration file, `named.conf`, will be in `/var/named/chroot/etc/named.conf` instead of `/etc/named.conf`. All other configuration and zone files will also be relative to this directory.

Each BIND server can manage one or more zones. Because of the time critical nature of DNS queries, it is imperative that name servers be available when clients request lookups. For this reason, each BIND server can also act as a backup, or *slave* other domains. In other words, a BIND server can be a master of one zone, and a slave for another. To avoid confusion, the *master zone files* are stored in the `/var/named/chroot/var/named` directory, and the *slave zone files* are stored in the `/var/named/chroot/var/named/slaves` directory.

When a slave server starts, it tries to contact the master and get a current copy of the database. The first time the master responds, it transfers the master zone file to the slave. The slave stores the zone file in the `slaves` directory, and awaits requests. Should the slave restart, it will contact the master and ask for the serial number of the current zone file. If the number is the same as the version in its store, it loads the existing file, thus reducing network traffic and server load. Periodically, the slave will poll the master, to ensure it has the latest zone information.



## BIND: named.conf

- Global options:

```
options{
    forwarders      { 203.50.0.137; };
    allow-query     { 192.168.0.0/24; };
    allow-transfers { 192.168.0.253; };
};
```

- Master zone:

```
zone "redhat.com" {
    type      master;
    file      "redhat.com.zone";
};
```

- Slave zone:

```
zone "kernel.org" {
    type      slave;
    masters   { 172.100.10.1; };
    file      "slaves/kernel.org.zone";
};
```

14-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Commonly used global option for named.conf are:

**forwarders** - Server forwards queries it can't answer to the name server at the IP addresses in this list. If it gets no answer, it will try a root name server unless the forward-only option is also set.

**allow-query** - Specifies an address match list of hosts allowed to query this server. If this option is not set, any host can query the server. In the example above, only hosts in the 192.168.0.0/24 subnet may query the DNS server.

**allow-transfer** - Like allow-query, specifies hosts that may copy the database. It should be used to limit zone transfers only to the slave servers.

Since a named.conf can not include host names, IP addresses will often be replaced with variables called *acl*'s. An *acl* assigns a name to a number, group of numbers, or a subnet, and make the file easier to read. They are completely optional.

The named.conf may also include a set of zone stanzas. Each zone statement will include the name of the zone and whether the server is mastering or slaving the zone. If the zone is a slave zone, it must have the IP address of the master. (This address could also be specified as an *acl*.) To avoid confusion,

ensure the slave files are sent to the `slave` directory. It is not unusual for forward and reverse lookup information to be maintained in two separate files. As a result, zone files often appear in pairs.

# BIND: Zone Files

- Begins with \$TTL
- Comments symbol is semicolon (;)
- Contains *resource records*
  - A - maps name to IP address
  - PTR - maps IP address to name
  - CNAME - creates an alias
  - MX - mail exchange record
- FQDN's must end with a dot.

14-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

All zone files start with a TTL directive. This determines the default length of time (in seconds) which a name server may cache the resource records. In certain advanced applications, this can be overridden on the individual resource records.

Names that appear in resource records may be abbreviated by specifying only the hostname. In such cases, the name inherits the domain from the \$ORIGIN variable, which is assigned by the corresponding zone directive in the named.conf. A hostname could be fully qualified by ending it with a dot.

Another abbreviation that can be used in a zone file is the @ symbol. This symbol indicates that the record references the machine that is hosting the **named** process. It tells BIND this it is referencing its own machine.

## *Examples of resource records*

```
@           IN  NS  ns1.redhat.com.  
redhat.com. IN  NS  ns2.redhat.com.
```

```
ns1.redhat.com. IN  A  192.100.100.1      ; NS needs an A record  
ns2           IN  A  192.100.100.2      ; FQDNs aren't required  
mail          IN  A  192.100.100.3
```

```
pop          IN  CNAME mail              ; alias pop to mail
```

```
@           IN  MX  5 mail.redhat.com.   ; MX needs an A record
```

```
3.100.100.192.IN-ADDR.ARPA. IN PTR mail ; IP address is backwards
```

## Securing Infrastructure Services

Daemon	Port	libwrap	SELinux
syslog	514	no	yes
yum	20,21,80	no	yes
dhcpd	67,68	no	yes
ntp	123	no	yes
named	53	no	yes

14-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## End of Unit 14

- Questions and Answers
- Summary
  - syslogd
  - yum, createrepo
  - dhcpd
  - ntp
  - named

14-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 14

## Enterprise Infrastructure

---

**Goal:** To implement Enterprise Infrastructure solutions.

**System Setup:** Throughout this lab, the hostnames and domain names that you use will be based on the IP address of your machine. Any time the lab refers to stationX, you should replace X with your station number. Any references to stationY will mean an unprivileged account on a classmate's system, or a remote station identified by your instructor.

Your `tcp_wrappers` configuration should include the following:

```
# cat /etc/hosts.allow
ALL:127.0.0.1 [::1]
sshd:192.168.0. [fe80::]/64
```

Your `iptables` firewall rules should include the following:

```
# cat /etc/sysconfig/iptables
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/255.255.255.0 -p tcp -m tcp
    --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --sport 53 -j ACCEPT
-A INPUT -s 192.168.0.254 -j ACCEPT
-A INPUT -j LOG
-A INPUT -j REJECT --reject-with icmp-port-unreachable
```

## Sequence 1: Logging to a centralized log host

Scenario: Your boss thinks it is a great idea to have one central logging host.

Work together with your neighbor. Set up one machine as the log server and the other as a log client

### Instructions:

1. First, on the log server set up syslogd to accept remote messages.
2. On the log client set up syslogd to send messages from the user facility to the log server.
3. Test the new setup by using logger to generate a syslog message. Does the message appear in your neighbor's `/var/log/messages`?

## Sequence 2: Working With BIND

**Scenario:** To expedite network transactions and decrease outbound bandwidth, your organization has elected to deploy a caching name server. Activate the service, using another in-house server as your forwarder.

**Deliverable:** A local name server for caching DNS transactions.

### Instructions:

1. Install the **caching-nameserver** and **bind-chroot** packages and start the **named** service. Modify `/etc/resolv.conf` to set **nameserver** to `127.0.0.1`. Execute **host station100**. Why did this command fail?
2. The install should have created an `/etc/named.caching-nameserver.conf` in the chrooted directory. In the **options** section add the line **forwarders** `{192.168.0.254; };` and restart the **named** service. Execute **host station100**. Why did this command work?
3. At this point, you have a working caching name server. Wasn't that difficult? Admittedly, there is one fairly large step left before we could put this into production. Execute the following command:

```
# grep "127.0.0.1\|localhost;" named.caching-nameserver.conf
```

You should see an output similar to the following:

```
listen-on port 53 { 127.0.0.1; };
allow-query      { localhost; };
match-clients    { localhost; };
match-destinations { localhost; };
```

These lines lock the caching name server to localhost operation. Each would have to be appended with network specific information to allow authorized clients to connect.



## Sequence 3: Migrating to a Zone Server

**Scenario:** To resolve local IP addresses, your organization has elected to deploy a local zone server. Since this is an in-house server rather than a public server, it will also need to be retrofitted for DNS caching.

**Deliverable:** A DNS server able to resolve local domain names.

**System Setup:**

**Instructions:**

1. Restart the name server, then tail the last twenty lines of `/var/log/messages`. You should see a line similar to the following:

```
Mar  4 18:02:36 stationX named[2305]: loading configuration from  
'/etc/named.caching-nameserver.conf'
```

The process is reading the configuration file for the caching name server.

2. In the chrooted `/var/named` directory, you will see a `name.ca` file. This file contains the addresses of the Root Name Servers that the caching name server uses, should the forwarder not provide a response. Make a copy of the file as `named.root`.

Install **system-config-bind**. Once installed, stop the `named` service. When you run `system-config-bind`, it will warn that it needs to create a default configuration file. Click OK.

*Note: If the training network does not have outside access, an error message will appear. Do not click OK. Close the window by clicking the [X] icon.*

Exit the utility. Look in the chrooted `/etc` directory. There should now be a `named.conf` file. Restart the `named` service. Which configuration file did it read?

3. Edit the new `named.conf` file. Examine the options section. You may notice there is no forwarder or security statements. This might be normal for a public name server that intends to server only "out of domain" queries against its own zones. In an enterprise environment, we may want the same server to do both tasks.

Add a forwarders line, save the change, and restart the service.

4. Launch `system-config-bind`. Click Properties. In the left hand pane, do you see forwarders? Highlight the entry. You should see server1's address in the Address List field.

In the right hand pane, highlight `allow-query`, and click `[+]` (the plus symbol) to move it to the left hand pane. In the *Edit List Element* fields, input `127.0.0.1` and click Add. This will allow your own system to query the name server. Create a second entry for `192.168.0.0/24`. Click OK, Save, and exit the application.

5. Examine the contents of your chrooted `/var/named` directory. Do you see any zone files? In order to act as a zone server, there will need to be a file for each domain.

6. Launch the **system-config-bind** application. Click the *New* icon, and select *Zone*. Under *Class*, click *OK* to select *Internet*, and under *Origin Type* click *OK* to select *Forward*. This will open a field for a new zone name. Create a zone name using your name followed by *.lan*. (that's "dot L A N dot") and click *OK*.
7. A new window will open. At the bottom of the window, will be a field asking for the *Zone File Path*. This will be the zone name, followed by *.db*. When you click *OK* an entry for your new domain will appear in the configuration window. Click *Save* and quit the configuration utility.
8. Again, list the chrooted */var/named* directory. This time you should see the newly created zone file. Review the contents of the file. The only entry in the file will reference your system as the name server.
9. Launch **system-config-bind**. Highlight the entry for you *.lan* domain. Click *New* and select *A IPv4 Address*. In the new window, create an entry for a host named *dragon*. For the IP address, add 100 to your system's address (example: station1 would use 192.168.0.101), then click *OK*. Exit **system-config-bind**, saving your changes.
10. Execute **host dragon.yourname.lan**: the command should fail. Try **host dragon.yourname.lan**. Why did this one work?
11. From the command prompt, load your zone file in an editor. Examine the entry for *dragon*. Add an additional entry for *poodle*, adding another 100 to *dragon*'s address (example: station1 would use 192.168.0.201). Save the file.
12. Execute **host poodle.yourname.lan**. The command should fail, as the **named** service was not reloaded to recognize the change. Issue the command **service named reload**, and try the command again.
13. Launch **system-config-bind** and expand your domain name. You should see an entry for *poodle*. Collapse your zone entry. There should also be an entry with an IP address of 192.168.0. This is the reverse lookup zone. Expand the reverse lookup zone. There should be an entry for *dragon*, but not for *poodle*. Close the application and return to the terminal.
14. In the same directory, you should find a *192.168.0.db* file. Load the file in an editor and add create an entry for *poodle* similar to the entry for *dragon* pointing to the appropriate IP address. Save the change and reload the server. Test your change with **host 192.168.0.2XX**.
15. Change to the chrooted etc directory and review the *named.conf* file. There should be master entries for your zone and the reverse lookup zone.

## Sequence 1 Solutions

1. First, on the log server set up syslogd to accept remote messages.

- a. Edit `/etc/sysconfig/syslog`:

```
SYSLOGD_OPTIONS="-r -m 0"
```

- b. Restart syslogd:

```
# service syslog restart
```

2. On the log client set up syslogd to send messages from the user facility to the log server.

- a. Append in `/etc/syslog.conf` the following line:

```
user.* @192.168.0.X
```

- b. Restart syslogd.

```
# service syslog restart
```

3. Test the new setup by using logger to generate a syslog message. Does the message appear in your neighbor's `/var/log/messages`?

```
# logger -i -t yourname "This is a test"
```

## Sequence 2 Solutions

1. Install the **caching-nameserver** and **bind-chroot** packages and start the **named** service.

```
# yum install -y caching-nameserver bind-chroot
# service named start
```

Modify `/etc/resolv.conf` to set nameserver to `127.0.0.1`. Execute **host station100**. Why did this command fail?

```
# cat /etc/resolv.conf
search example.com
nameserver 127.0.0.1
# host station100
Host station100 not found: 3(NXDOMAIN)
```

When issuing a host request, remember that unless the name is fully qualified, the search argument from `/etc/resolv.conf` is appended to the query. In this case, we are actually querying for `station100.example.com`. You are not getting a response because, by default, a caching name server only communicates with the Root Name Servers. The host name, `station100` is only on `server1`, thus can not be resolved.

2. The install should have created an `/etc/named.caching-nameserver.conf` in the chrooted directory. In the options section add the line: `forwarders {192.168.0.254;};`

```
# cd /var/named/chroot/etc
# grep 192.168.0.254 named.caching-nameserver.conf
forwarders {192.168.0.254;};
```

Restart the **named** service. Execute **host station100**.

```
# service named restart
# host station100
station100.example.com has address 192.168.0.100
```

In this case, `server1` was the first stop for host name lookups. Had `server1` not provided the answer, the query would have been forwarded to the Root Name Servers.

3. At this point, we have a working caching name server. Wasn't that difficult?

No

Admittedly, there is one fairly large step left before we could put this into production. Execute the following command:

```
# grep "127.0.0.1|localhost;" named.caching-nameserver.conf
```

You should see an output similar to the following:

```
listen-on port 53 { 127.0.0.1; };
allow-query { localhost;};
```

```
match-clients      { localhost; };  
match-destinations { localhost; };
```

These lines lock the caching name server to localhost operation. Each would have to be appended with network specific information to allow authorized clients to connect.

## Sequence 3 Solutions

1. Restart the name server, then tail the last twenty lines of /var/log/messages. Your results should be similar to the following:

```
# service named restart
# tail -20 /var/log/messages | grep loading
Mar  4 18:02:36 stationX named[2305]: loading configuration from ✓
'/etc/named.caching-nameserver.conf'
```

The process is reading the configuration file for the caching name server. In order to migrate from caching name server to zone server, we need a new configuration file. We could modify the existing file or create another, but there maybe an easier way.

2. In the chrooted /var/named directory, you will see a name .ca file. This file contains the addresses of the Root Name Servers that the caching name server uses, should the forwarder not provide a response. Make a copy of the file as named .root.

```
# cd /var/named/chroot/var/named
# cp named.ca named.root
```

Install **system-config-bind**.

```
# yum install -y system-config-bind
Installed: system-config-bind.noarch 0:4.0.3-2.el5
Complete!
```

*Once installed, stop the named service.*

```
# service named stop
```

When you run system-config-bind, it will warn that it needs to create a default configuration file. Click OK.

*Note: If the training network does not have outside access, an error message will appear. Do not click OK. Close the window by clicking the [X] icon.*

Exit the utility. Look in the chrooted /etc directory. There should now be a named.conf file. Start the named service. Which configuration file did it read?

```
# service named start
Starting named: [ OK ]
[root@stationX etc]# tail -20 /var/log/messages | grep loading
Mar  4 20:10:39 stationX named[5647]: loading configuration from ✓
'/etc/named.conf'
```

3. Edit the new named.conf file. Examine the options section. You may notice there is no forwarder or security statements. This might be normal for a public name server that intends to server only "out of domain" queries against its own zones. In an enterprise environment, we may want the same server to do both tasks.

Add a forwarders line, save the change, and restart the service.

```
# vi named.conf
# grep "forwarders" named.conf
    forwarders { 192.168.0.254; };
# service named restart
Stopping named: [ OK ]
Starting named: [ OK ]
```

4. Launch **system-config-bind**. Click **Properties**. In the left hand pane, do you see forwarders? Highlight the entry. You should see server1's address in the Address List field.

In the right hand pane, highlight **allow-query**, and click **[+]** (the plus symbol) to move it to the left hand pane. In the *Edit List Element* fields, input **127.0.0.1** and click **Add**. This will allow your own system to query the name server. Create a second entry for **192.168.0.0/24**. Click **OK**, **Save**, and exit the application.

5. Examine the contents of your chrooted **/var/named** directory. Do you see any zone files?

```
# cd -
/var/named/chroot/var/named
# ls *.zone *.db
ls: *.db: No such file or directory
localdomain.zone  localhost.zone
```

In order to act as a zone server, there will need to be a file for each domain.

6. Launch the **system-config-bind** application. Click the **New** icon, and select **Zone**. Under **Class**, click **OK** to select **Internet**, and under **Origin Type** click **OK** to select **Forward**. This will open a field for a new zone name. Create a zone name using your name followed by **.lan**. (that's "dot L A N dot") and click **OK**.

7. A new window will open. At the bottom of the window, will be a field asking for the **Zone File Path**. This will be the zone name, followed by **.db**. When you click **OK** an entry for your new domain will appear in the configuration window. Click **Save** and quit the configuration utility.

8. Again, list the chrooted **/var/named** directory. This time you should see the newly created zone file.

```
# ls *.zone *.db
localdomain.zone  localhost.zone  yourname.lan.db
```

Review the contents of the file. The only entry in the file will reference your system as the name server.

```
# cat yourname.lan.db
$TTL 1H
@ SOA stationX.example.com. root.stationX.example.com. ( 2
... output truncated...
```

9. Launch **system-config-bind**. Highlight the entry for your *.lan* domain. Click *New* and select *A IPv4 Address*. In the new window, create an entry for a host named *dragon*. For the IP address, add 100 to your system's address (example: station1 would use 192.168.0.101), then click *OK*. Exit **system-config-bind**, saving your changes.

10. Execute **host dragon**: the command should fail.

```
# host dragon
Host dragon not found: 3 (NXDOMAIN)
```

Try **host dragon.yourname.lan**. Why did this one work?

```
# host dragon.yourname.lan
dragon.yourname.lan has address 192.168.0.119
```

The host command appends resolv.conf's search directive to the file. Thus the first query was executed against dragon.example.com. Shouldn't that have been to server1? Glad you asked. No. When you launched system-config-bind, it did not migrate your caching name server configuration, it created a default configuration.

11. From the command prompt, load your zone file in an editor. Examine the entry for *dragon*. Add an additional entry for *poodle*, adding another 100 to *dragon*'s address (example: station1 would use 192.168.0.201). Save the file.

```
# vi student.lan.db
# grep "192.168.0" student.lan.db
dragon IN      1H      A      192.168.0.1XX
poodle IN      1H      A      192.168.0.2XX
```

12. Execute **host poodle.yourname.lan**. The command should fail.

```
# host poodle.yourname.lan
Host poodle.yourname.lan not found: 3 (NXDOMAIN)
```

Since the **named** service was not reloaded, the server did not recognize the change. Issue the command **service named reload**, and try the command again. In the previous examples the application handled the reload for you.

```
# service named reload
Reloading named: [ OK ]
# host poodle.yourname.lan
poodle.yourname.lan has address 192.168.0.2XX
```

13. Launch system-config-bind and expand your domain name. You should see an entry for poodle. Collapse your zone entry. There should also be an entry with an IP address of 192.168.0. This is the reverse lookup zone. Expand the reverse lookup zone. There should be an entry for dragon, but not for poodle. Close the application and return to the terminal.



14. In the same directory, you should find a 192.168.0.db file. Load the file in an editor and add create an entry for poodle similar to the entry for dragon pointing to the appropriate IP address. Save the change and reload the server.

```
# vi 192.168.0.db
# grep "yourname.lan" 192.168.0.db
1XX PTR dragon.yourname.lan.
2XX PTR poodle.yourname.lan.
# service named reload
```

Reloading named:

[ OK ]

Test your change with host 192.168.0.2XX.

```
# host 192.168.0.2XX
2XX.0.168.192.in-addr.arpa domain name pointer
poodle.yourname.lan.
```

15. Change to the chrooted etc directory and review the named.conf file. There should be master entries for your zone and the reverse lookup zone.

# Unit 15

## HTTP Service

15-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Understand the capabilities of Apache
- Configure Apache
- Secure the Apache service
- Set up virtual hosting
- Configure the Squid proxy server

15-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Apache Overview

- Process control:
  - spawn processes before needed
  - adapt number of processes to demand
- Dynamic module loading:
  - run-time extensibility without recompiling
- Virtual hosts:
  - Multiple web sites may share the same web server

15-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Apache server has a flexible mechanism for accepting requests and dispatching children to process them which is abstracted into *Multi-Processing Modules* (MPM). The MPM used by default in Red Hat Enterprise Linux (RHEL) is *prefork*, which spawns multiple child processes when needed just like Apache 1.3. Other MPMs are not yet available, although directives for some appear in the configuration file.

Dynamic module loading allows a web server administrator to change the behavior of Apache. This can be done without recompiling any source code, and simply specifying the use of a given module. An example of a commonly used module is `mod_perl`, used to increase Perl CGI script execution speed.

The Apache HTTP Server project web site is <http://httpd.apache.org>.

# Apache Security

- Listens on 80/tcp, optionally 443/tcp
- **/usr/sbin/httpd** does not use **tcp\_wrappers**
- Confined by SELinux as **http\_t**
- Runs as user **apache**, group **apache** and must have a appropriate permissions to access directories and files
- Uses internal **Allow** and **Deny** statements

15-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The Apache server operates on port 80/tcp, though this is configurable. If the **mod\_ssl** has been loaded, the server will also listen on port 443/tcp (configurable.)

```
[root@stationX ~]# nmap 192.168.0.X
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-01-31 22:10 EST
Interesting ports on 192.168.0.X:
Not shown: 1677 closed ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
Nmap finished: 1 IP address (1 host up) scanned in 0.139 seconds
```

The server is run as the **/usr/sbin/httpd**. It does not implement **libwrap.so**.

```
[root@stationX ~]# ldd /usr/sbin/httpd | grep libwrap.so
[root@stationX ~]# strings /usr/sbin/httpd | grep "hosts\..."
```

The SELinux configuration for a Apache is perhaps the single most extensive of all services. There are about 13 different security context types. Some types of importance:

- **httpd\_sys\_content\_t** = web content
- **httpd\_sys\_script\_exec\_t** = scripts
- **httpd\_log\_t** = log files

There are also about 16 booleans. Booleans of importance:

- **httpd\_disable\_trans** = disable SELinux restriction of Apache
- **httpd\_enable\_cgi** = allow Apache to execute CGI scripts
- **httpd\_enable\_ftp\_server** = allow Apache to act as an FTP server
- **httpd\_enable\_homedirs** = allow Apache access home directories

# Apache Server Configuration

- Configuration Files:
  - /etc/httpd/conf/httpd.conf
  - /etc/httpd/conf.d
  - content directories via *overrides*
- Global Parameters
  - `ServerRoot` - chroot for relative files
  - `StartServers` - number of httpd process to start
  - `MaxClients` - simultaneous request limit  $\leq$  `ServerLimit`
  - `Modules` - shared objects to load at start
  - `Include` - include other configs (`conf.d`)
- Main / Virtual Parameters
  - `DocumentRoot` - where to find content files
  - `HostnameLookups` - resolve IPs to names for logging
  - `ErrorLog` - where to log server errors
  - `CustomLog` - where to log requests

15-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The files used by Apache are potentially spread across the system, but `/etc/httpd/`, the `ServerRoot`, contains the most important. In `/etc/httpd/` you should see a few symbolic links that point to the location of Apache's log files and dynamically loadable modules include by `LoadModule` directives.

Parameters are divided into two categories: *global* and *main*. In the configuration file, they appear under three categories, including *virtual*. By default only the global and main sections are used, and the virtual section is disabled. If virtual hosting is switched on, the virtual hosts inherit the settings of global and main. It is possible for a virtual host to override an directive in the main section, but not the global section.

## Creating an Alternate DocumentRoot

- Create a directory as group apache with 2750 permissions
  - `chmod g+s,o-rx`
- Change security context to `httpd_sys_content_t`
- Create a content subdirectory with 2770 permissions
  - `chmod g+w,g+s,o-rx`
- Add users to the apache group
- Change DocumentRoot and reload the server

15-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

By default only root can modify web content on an Apache server. This is a less than desirable situation. As a best practice, web content administrators should be put in a unique group, and content should be put in a limited access directory. Since several layers of security surround the Apache server, DocumentRoot can not be casually changed. Consider the following creation of a collaborative web directory:

```
[root@stationX ~]# mkdir /www
[root@stationX ~]# ls -ld /www
drwxr-xr-x 2 root root 4096 Jan 31 23:31 /www
[root@stationX ~]# chgrp apache /www
[root@stationX ~]# chmod 2750 /www
[root@stationX ~]# ls -ld /www
drwxr-s--- 2 root apache 4096 Jan 31 23:31 /www
```

The directory is readable by the apache group, but not writable. However, SELinux will restrict access because of an improper security context.

```
[root@stationX ~]# ls -Zd /www
drwxr-s--- root root root:object_r:root_t /www
[root@stationX ~]# chcon --reference /var/www/html /www
[root@stationX ~]# ls -Zd /www
drwxr-s--- root apache system_u:object_r:httpd_sys_content_t /www
```

The directory is now accessible to both users in the apache group and the **httpd** process. This directory will be the parent, with content below.

```
[root@stationX ~]# mkdir /www/html
[root@stationX ~]# chmod 2770 /www/html
[root@stationX ~]# ls -Zd /www/html
drwxrwsr-x root apache root:object_r:httpd_sys_content_t /www/html
```

The new directory inherited its parent's security context. With the **chmod 2770** statement the subdirectory was made writable by the group, and all files and subdirectories will inherit the group affiliation. At this point, the content creators can get to work.

It may be helpful to create a `/www/cgi-bin` for scripts and a `/www/log` for logs. Make sure both are group writable.



# Virtual Host Example

- Global configuration

```
NameVirtualHost *:80
# Default virtual host
<VirtualHost *:80>
    ServerName localhost
</VirtualHost>
```

- Virtual Host configuration

```
<VirtualHost *:80>
    ServerName     example.com
    ServerAlias    www.example.com
    DocumentRoot  /var/www/vhost/example.com/html
</VirtualHost>
```

15-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The virtual host sections of `httpd.conf` usually have more directives than just `ServerName`, and `DocumentRoot`. These are the absolute minimum. Other likely directives include `ErrorLog`, `CustomLog`, and `ScriptAliases`.

Once any virtual hosts are defined, all content served from the server must be moved into a virtual host. Consider the following example of a "main" server, under default operation:

```
[root@stationX ~]# links --dump http://192.168.0.X
Main DocumentRoot = /var/www/html
```

With virtual hosts enabled:

```
[root@stationX ~]# links --dump http://example.com
Virtual DocumentRoot = /www/html
[root@stationX ~]# links --dump http://192.168.0.X
Virtual DocumentRoot = /www/html
```

Since main server is disabled, and all requests return the virtual server's page. By adding an entry for `localhost` (as seen on the slide), we can regain the main server:

```
[root@stationX ~]# links --dump http://example.com
Virtual DocumentRoot = /www/html
[root@stationX ~]# links --dump http://192.168.0.19
Main DocumentRoot = /var/www/html
```

The order of the `VirtualHost` directives is important. The first `VirtualHost` becomes the default if no matching `ServerName` matches. In the example a below, if a browser tries to connect to

www.virt2.com, the content from /virt1 would be displayed, because www.virt2.com is not defined as a ServerName. To remedy this, add a ServerAlias line that defines other aliases:

```
<VirtualHost *:80>
  ServerName    virt1.com
  DocumentRoot  /virt1
</VirtualHost>
<VirtualHost *:80>
  ServerName    virt2.com
  DocumentRoot  /virt2
  ServerAlias   www.virt2.com www2.virt2.com
</VirtualHost>
```

As long as the DNS entries point to the defined IP address, www.virt2.com and www2.virt2.com will display the content from /virt2/

# Apache httpd Access Control Example

```
<Directory /var/www/vhost/example.com/html>
  # Allow is default and overrides deny
  Order deny,allow
</Directory>

<Directory /var/www/vhost/example.com/html/private>
  # Deny is default and overrides allow
  Order allow,deny
  # Allow a trusted subnet
  Allow from 192.168.0.
  # Override Allow to block an untrusted host
  Deny from 192.168.0.199
</Directory>
```

15-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `Directory` directive groups options that apply to files that would be served from the specified directory.

The `Order` directive controls the order that `Allow` and `Deny` directives are evaluated and sets the default if there is no match either for `Allow` or `Deny`.

For `Order allow,deny` the default is to deny access. `Allow` can allow some hosts to access the content. `Deny` can be used in the second path to deny hosts that would have been permitted by the `Allow`.

For `Order deny,allow` the default is to allow access. `Deny` can be used to block access to some hosts and `Allow` can override the `Deny`.

# Squid Web Proxy Cache

- Squid supports caching of FTP, HTTP, and other data streams
- Squid will forward SSL requests directly to origin servers or to one other proxy
- Squid includes advanced features including access control lists, cache hierarchies, and HTTP server acceleration

15-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Squid is an internet object cache that can act as a proxy server for HTTP, FTP, and other requests. Clients request URLs from Squid, which then either serves cached copies of the URLs if they have been previously requested. URLs associated with dynamic content (CGI executables, server-parsed pages) get forwarded, rather than being served out of the cache.

Squid may be used as an HTTP accelerator. Just as Squid makes URL requests on behalf of a client when it acts as a proxy, **squid** serves URL requests on behalf of a server when acting as an accelerator. For example, a site whose URL is `http://www.notreallyhttpd.com` may actually have a **squid** process listening for requests on port 80 of `www.notreallyhttpd.com`. It will either serve the page itself out of cache, or else it will forward the request to the web server that handles that site.

Squid's configuration file is `/etc/squid/squid.conf`, and the `squid` RPM includes the Squid project's well-commented example configuration file. Key configuration elements include the port number on which it will listen for requests, whether it is inside a firewall, timeout settings, and ICP request port number. Squid uses port 3128 by default, but can easily be changed to 8080 if required.

Like Sendmail, the default Squid configuration only accepts connections on the system's loopback interface. To allow local network access, add an `acl` directive corresponding to the local network (for example, `acl mynet src 192.168.0.0/255.255.255.0`) and add an `http_access` directive corresponding to the new `acl` before the line `http_access deny all`.

The ICP (Internet Cache Protocol) request port number relates to Squid's ability to participate in cache hierarchies. A Squid cache can share the contents of its cache, or can request URLs from the cache of other squid processes if it belongs to a cache hierarchy. A Squid cache may act as a parent, sibling, or child of another Squid cache. The default maximum size of the cache is 100 MB, which would need to be increased (using the `cache_dir` directive) for most realistic situations.

## Useful parameters in `/etc/squid/squid.conf`

- `http_port 3128`
- `cache_mem 8 MB`
- `cache_dir ufs /var/spool/squid 100 16 256`
- `acl all src 0.0.0.0/0.0.0.0`
- `acl localhost src 127.0.0.1/255.255.255.255`
- `http_access allow localhost`
- `http_access deny all`

15-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The `squid.conf` file is full of useful information. Some useful parameters contained in `squid.conf` are shown above with their default settings.

Note that the file is parsed from top to bottom. Thus, if you are attempting to allow other `acls`, place the `http_access` line *before* the `http_access deny all` entry.

## End of Unit 15

- Questions and Answers
- Summary
  - httpd
  - VirtualHost
  - squid

15-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 15

## HTTP Services

---

**Goal:** Install a basic web server with virtual host capabilities.

Configure a simple proxy server.

**System Setup:** Throughout this lab, the hostnames and domain names that you use will be based on the IP address of your machine. Any time the lab refers to stationX, you should replace X with your station number. Any references to stationY will mean an unprivileged account on a classmate's system, or a remote station identified by your instructor.

Your `tcp_wrappers` configuration should include the following:

```
# cat /etc/hosts.allow
ALL:127.0.0.1 [::1]
sshd:192.168.0. [fe80::]/64
```

Your `iptables` firewall rules should include the following:

```
# cat /etc/sysconfig/iptables
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp --sport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/255.255.255.0 -p tcp -m tcp
    --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --sport 53 -j ACCEPT
-A INPUT -s 192.168.0.254 -j ACCEPT
-A INPUT -j LOG
-A INPUT -j REJECT --reject-with icmp-port-unreachable
```

Your `/etc/resolv.conf` should be set as follows:

```
search example.com
nameserver 192.168.0.254
```

Test as follows:

```
# dig +short stationX.example.com
192.168.0.X
# dig +short wwwX.example.com
stationX.example.com.
192.168.0.X
```

## Sequence 1: Apache installation and configuration

Scenario: Build a simple Apache deployment for a single domain, with iptables security.

Deliverable: Remote access to `http://wwwX.example.com`.

### Instructions:

1. Stop **iptables** and install **httpd** from the yum server. Verify `DocumentRoot` is set to `/var/www/html` and start the service. Ensure the service will launch at boot time.
2. Use **links --dump** to test `http://localhost`. This should output a test page. (This page is not a real HTML document, but is generated by the server if a default page does not exist.) Create an `index.html` document in `DocumentRoot` as follows:  
  

```
# echo wwwX.example.com > /var/www/html/index.html
```

  
Without restarting or reloading the server, test the connection again.
3. Launch a new window to monitor `/var/log/messages`. Start **iptables** and attempt to access your server from `stationY`. If your firewall is configured correctly, this should fail.
4. Review the log entry in your monitoring window and construct an **iptables** rule to allow the connection. Use **links** to verify.



## Sequence 2: Migrating to a Virtual Web server

**Scenario:** A couple of web development teams have been putting together web sites that will need to be deployed on the same server. Convert your current server to support virtual hosting.

**Deliverable:** A working web server supporting both `wwwX.example.com` and `stationX.example.com`

### Instructions:

1. Create the virtual server's directory: `/srv/stationX`. Modify `httpd.conf` to enable virtual hosting. Add a `VirtualHost` stanza at the very bottom of the config file using the following settings:  
  

```
Server Name:          stationX.example.com
Document Directory:   /srv/stationX/www
Error Log:            /var/log/httpd/stationX_error_log
Custom Access Log:    /var/log/httpd/stationX_access_log
```

common
2. Restart the server to implement the changes. Launch a monitoring window to watch both `/var/log/messages` and the server's logs. Use **links --dump** to test the `http://stationX.example.com` virtual host. It should fail. Determine the nature of the failure and correct the problem.
3. Create a `/srv/stationX/www/index.html` file containing *stationX.example.com*. Use **links --dump** to test the `http://stationX.example.com` virtual host. It should fail, instead showing the test message. Try to view the file as follows:  
  

```
links --dump http://stationX.example.com/index.html
```

This should yield a different error message. Use the error logs to troubleshoot the problem and try again.
4. Now that the virtual host is working, use **links --dump** to test the main server configuration from the previous exercise. It should show the wrong page. Try `http://localhost`. Once virtual hosting is made active, the main server is no longer accessible. Extend the virtual host configuration to fix this problem.
5. At this point, we've met our objective, and should probably leave well enough alone, but being intrepid explorers, we shall go one step further. Presently, `http://192.168.0.X` returns the `wwwX.example.com` page. Change the configuration such that it returns the `stationX.example.com`. After all, it makes sense that your hostname and IP address would return the same page.

## Sequence 3: Basic Squid configuration

Deliverable: A proxy server

### Instructions:

1. Install squid on your system. Ensure the service is set to launch at boot time.
2. Start the **squid** service, then configure your web browser to use your proxy. Use localhost as the server, with the port set to 3128.
3. Try accessing a web page somewhere. If the classroom does not have Internet access, try `http://server1.example.com`, which should return the Apache test page.
4. Now have a neighbor configure his or her web browser to use your proxy. This should not work.
5. Configure the firewall to allow the default proxy port to your neighbor's IP address. Have your neighbor attempt to connect again. This should still not work.
6. The page that **squid** returns, and the bottom of `/var/log/squid/access.log` indicate why.
7. Open `/etc/squid/squid.conf` in your preferred text browser. As you can see, it is mostly comments and documentation. You will also note that squid is extremely configurable and tunable. For this lab, we will configure a basic setup that will be adequate for many settings.
8. Search for the second occurrence of Recommended minimum configuration in the file. This will take you to the default access control lists, or `acls`. Add an `acl` for the `192.168.0.0/24` network below the `acl CONNECT method CONNECT` line.
9. Search further down in the file for `INSERT YOUR OWN RULE(S) HERE`. Add a line above the `localhost` access rule to allow `example`.

Reload **squid**. Your neighbor should now be able to access your cache.

10. Some URLs are best avoided completely. Return to the `acl` section, and add the some `acl` lines beneath the line you added earlier to include `.yahoo.com` and `.hotmail.com` in an `acl` named `otherguys` (use `.example.com` if you do not have Internet access in your classroom).
11. Add a rule to deny access to these problematic domains.

Restart **squid** again, then check one or more of the web sites associated with those domains. Unfortunately, you find that access is not denied.

12. Open the configuration file again, and move the `deny` rule you added so that it is before the `allow` rule for `example`. Order matters, so by having the `allow` rule for `example` before the `deny` rule for the `otherguys` destinations, access was allowed and the `deny` rule never took effect. After moving the rule, restart **squid** once more. This time it should deny access to any site within the prohibited domains.

## Sequence 1 Solutions

1. Stop **iptables** and install **httpd** from the yum server.

```
# service iptables stop
# yum install -y httpd
... output truncated ...
Installed: httpd 0:2.2.3-6.el5
Dependency Installed: apr 0:1.2.7-10 apr-util 0:1.2.7-3 ✓
    postgresql-libs 0:8.1.4-1.1
Complete!
```

Verify DocumentRoot is set to /var/www/html and start the service. Ensure the service will launch at boot time.

```
# grep -i "^DocumentRoot" /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
# service httpd status
httpd is stopped
# chkconfig httpd --list
httpd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
# service httpd start
Starting httpd:                                     [ OK ]
# chkconfig httpd on
```

2. Use **links --dump** to test `http://localhost`.

```
# links --dump http://localhost | head -1
Red Hat Enterprise Linux Test Page
```

This page is not a real HTML document, but is generated by the server if a default page does not exist. Create an `index.html` document in DocumentRoot as follows:

```
# echo wwwX.example.com > /var/www/html/index.html
```

Without restarting or reloading the server, test the connection again.

```
# links --dump http://localhost | head -1
wwwX.example.com
```

3. Launch a new window to monitor /var/log/messages.

```
# xterm -e "tail -f /var/log/messages" &
```

Start **iptables**.

```
# service iptables start
Applying iptables firewall rules:                    [ OK ]
```

Attempt to access your server from stationY. If your firewall is configured correctly, this should fail.

```
[stationY]$ links --dump http://wwwX.example.com | head -1
Elinks: Connection refused
```

4. In your monitoring window should be an entry similar to the following:

```
Feb 26 19:22:06 stationX kernel: IN=eth0 OUT=
MAC=00:16:3e:03:10:03:0d:60:fa:f5:f2:08:00 SRC=192.168.69.32
DST=192.168.0.X LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=20564 DF
PROTO=TCP SPT=53899 DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
```

Near the end of the line notice the PROTO and DPT entries. These indicate that the rejected packet was protocol TCP and was destined for port 80. With this information, we could build a simple rule to allow connections.

```
# iptables -I INPUT 5 -p tcp --dport 80 -j ACCEPT
service iptables save
```

Attempt to connect to the web server from stationY.

```
[stationY]$ links --dump http://wwwX.example.com | head -1
wwwX.example.com
```

Ideally, we would tweak the **iptables** rule to only allow friendly clients. Alternately, we could block unwanted clients by adding DROP rules *before* the ACCEPT rule.

## Sequence 2 Solutions

1. Create the virtual server's directory: `/srv/stationX`.

```
# mkdir /srv/stationX
```

Modify the server's `httpd.conf` to enable virtual hosting. Add a `VirtualHost` stanza at the very bottom of the config file using the following settings:

```
Server Name:          stationX.example.com
Document Directory:   /virtual/stationX/www
Error Log:            /var/log/httpd/stationX_error_log
Custom Access Log:    /var/log/httpd/stationX_access_log common
```

To enable virtual hosting, the first step is to configure the `NameVirtualHost` line with an interface argument. By using `*:80`, we ensure the server will listen on all interfaces. Next we create an opening and closing `VirtualHost` directive. These directives must include the same interface argument as the `NameVirtualHost` line.

At a minimum, we need a `ServerName` and `DocumentRoot` directive. It's best to also include an `ErrorLog` and `CustomLog` directive to separate each hosts log messages.

```
# vi /etc/httpd/conf/httpd.conf
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName stationX.example.com
    DocumentRoot /srv/stationX/www
    ErrorLog /var/log/httpd/stationX_error_log
    CustomLog /var/log/httpd/stationX_access_log common
</VirtualHost>
```

2. Restart the server to implement the changes. Launch a monitoring window to watch both `/var/log/messages` and the server's logs.

```
# service httpd restart
# xterm -e "cd /var/log; tail -f messages httpd/*" &
```

Use **links --dump** to test the `http://stationX.example.com` virtual host. It should fail.

```
# links --dump http://stationX.example.com | head -1
Not Found
```

Look at the error logs, there should be a message similar to the following:

```
==> httpd/wwwX_error_log <==
[Tue Feb 27 23:05:34 2007] [error] [client 192.168.0.X] File
does not exist: /srv/stationX/www
```

In the first step we created `/srv/stationX`, but did not create the underlying file structure to support the html page. Create the appropriate sub-directory, and try again.

```
# mkdir -p /srv/stationX/www
# links --dump http://stationX.example.com | head -1
Red Hat Enterprise Linux Test Page
```

3. Create a `/srv/stationX/www/index.html` file containing *stationX.example.com*. Use **links --dump** to test the `http://stationX.example.com` virtual host. It should fail, instead showing the test message.

```
# echo stationX.example.com > /srv/stationX/www/index.html
# links --dump http://stationX.example.com | head -1
Red Hat Enterprise Linux Test Page
```

Try to view the file appending the filename to the previous URI. This should yield a different error message.

```
# links --dump http://stationX.example.com/index.html | head -1
Forbidden
```

In the error logs there should be a message similar to the following:

```
Feb 27 23:49:44 stationX setroubleshoot: SELinux is
preventing the /usr/sbin/httpd from using potentially mislabeled
files /srv/stationX/www/index.html (var_t). For complete
SELinux messages. run sealert -l
9831b35b-79b0-44fa-aa62-b31ef56ab050
```

The message is indicating that the **httpd** process was restricted from accessing `index.html`. View the SELinux security context of `index.html` and the security context of the `index.html` from the previous exercise.

```
# ls -Z /srv/stationX/www/index.html
-rw-r--r-- root root root:object_r:var_t
/srv/stationX/www/index.html
# ls -Z /var/www/html/index.html
-rw-r--r-- root root root:object_r:httpd_sys_content_t
/var/www/html/index.html
```

Notice that the type element (`_t`) of the security context is different. We could use **chcon** to force the context, but there maybe an easier way. Try to **restorecon -R** on the parent directory.

```
# restorecon -R /srv/stationX/www
# ls -Z /var/www/html/index.html
-rw-r--r-- root root root:object_r:httpd_sys_content_t
/var/www/html/index.html
# links --dump http://stationX.example.com/index.html | head -1
stationX.example.com
```

Since we are using a pre-approved directory, **restorecon** knows the correct context and will correct the problem. (Pre-approved directories include `/srv/*/www`,

/home/\*/public\_html, and /var/www/html. Others can be specified: see **man semanage**.)

4. Now that the virtual host is working, use **links --dump** to test the main server configuration from the previous exercise. It should fail. Try `http://localhost`.

```
# links --dump http://wwwX.example.com/index.html | head -1
stationX.example.com
# links --dump http://localhost | head -1
stationX.example.com
```

Once virtual hosting is made active, the main server is no longer accessible. The virtual host configuration will have to be extended to fix this problem. Edit the `httpd.conf` and insert the following lines immediately after `NameVirtualHost`:

```
<VirtualHost *:80>
    ServerName localhost
</VirtualHost>
```

Reload the server and try to access `http://localhost`.

```
# links --dump http://localhost | head -1
wwwX.example.com
# links --dump http://wwwX.example.com | head -1
wwwX.example.com
```

As an alternative, you could have created another complete virtual host stanza for `wwwX.example.com`.

5. At this point, we've met our objective, and should probably leave well enough alone, but being intrepid explorers, we shall go one step further. Presently, `http://192.168.0.X` returns the `wwwX.example.com` page. Change the configuration such that it returns the `stationX.example.com`. After all, it makes sense that your hostname and IP address would return the same page.

The hard way of doing this is to construct another virtual host stanza for the IP address. The easy solution is a `ServerAlias`. Edit the `httpd.conf` and immediately after the `ServerName stationX.example.com` line, add the following:

```
ServerAlias 192.168.0.X
```

Reload the server and test.

```
# service httpd reload
Reloading httpd: [ OK ]
# links --dump http://192.168.0.X | head -1
stationX.example.com
```

## Sequence 3 Solutions

1. Install squid on your system. Ensure the service is set to launch at boot time.  

```
# yum install -y squid  
# chkconfig squid on
```
2. Start the **squid** service, use **chkconfig** to enable **squid** at boot, then configure your web browser to use your proxy. Use **localhost** as the server, with the port set to 3128.
  - a. 

```
# service squid start
```
  - b. To set the proxy settings in Firefox, navigate to Edit->Preferences. In the **General** settings, click on the **Connection Settings...** button. Click the **Manual proxy configuration** radio button. Add **localhost** in the **HTTP Proxy:** box, and 3128 in the **Port:** box. Click **OK** to accept the changes.
3. Try accessing a web page somewhere. If the classroom does not have Internet access, try `http://server1.example.com`, which should return the Apache test page.
4. Now have a neighbor configure his or her web browser to use your proxy. This should not work.
5. Configure the firewall to allow the default proxy port to your neighbor's IP address. Have your neighbor attempt to connect again. This should still not work.
  - a. 

```
# iptables -I INPUT 2 -p tcp --dport 3128 -j ACCEPT
```
  - b. 

```
# service iptables save
```
6. The page that **squid** returns, and the bottom of `/var/log/squid/access.log` indicate why.
7. Open `/etc/squid/squid.conf` in your preferred text browser. As you can see, it is mostly comments and documentation. You will also note that squid is extremely configurable and tunable. For this lab, we will configure a basic setup that will be adequate for many settings.
  - a. If you don't have a preferred text editor, use **gedit**:  

```
# gedit /etc/squid/squid.conf
```
8. Search for the second occurrence of **Recommended minimum configuration** in the file. This will take you to the default access control lists, or **acls**. Add an **acl** for the 192.168.0.0/24 network below the **acl CONNECT method CONNECT** line.
  - a. You should add this line:  

```
acl example src 192.168.0.0/24
```

You can now refer to this network as **example** elsewhere in the configuration file. **src** means that the IP specified is the source IP(s) for this **acl**.



9. Search further down in the file for **INSERT YOUR OWN RULE(S) HERE**. Add a line above the `localhost` access rule to allow `example`.

Reload **squid**. Your neighbor should now be able to access your cache.

- a. You should add this line:

```
http_access allow example
```

- b. **# service squid reload**

10. Some URLs are best avoided completely. Return to the `acl` section, and add the some `acl` lines beneath the line you added earlier to include `.yahoo.com` and `.hotmail.com` in an `acl` named `otherguys` (use `.example.com` if you do not have Internet access in your classroom).

- a. Add these lines near the location that you added the `example` `acls`:

```
acl otherguys dstdomain .yahoo.com
acl otherguys dstdomain .hotmail.com
```

- b. There are a couple of things to mention here. First, note that the additive property of `acls`. Both of the domains are added to the `acl`. Second, note the `dstdomain` `acl` type, which specifies that this definition concerns destination domains. Third, note the use of dot notation in specifying the domain name. Make sure to include the leading dot.

11. Add a rule to deny access to these problematic domains.

Restart **squid** again, then check one or more of the web sites associated with those domains. Unfortunately, you find that access is not denied.

- a. Return to where you added the allow rule for `example`, and below it add the following:

```
http_access deny otherguys
```

- b. **# service squid reload**

12. Open the configuration file again, and move the deny rule you added so that it is before the allow rule for `example`. Order matters, so by having the allow rule for `example` before the deny rule for the `otherguys` destinations, access was allowed and the deny rule never took effect. After moving the rule, restart **squid** once more. This time it should deny access to any site within the prohibited domains.

- a. **# service squid reload**

# Unit 16

## Mail Service

16-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## Objectives

Upon completion of this unit, you should be able to:

- Understand electronic mail operation
- Configure a Sendmail server
- Configure a Postfix server
- Understand mail aliases
- Configure a Dovecot server

16-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

## An Email Review

- *Mail user agent (MUA)* passes message to *mail transport agent (MTA)*
- MTA routes message to destination, giving to other intermediate MTAs as necessary
- Domain MTA passes message to *mail delivery agent (MDA)*
- User receives message
  - Local spool in `/var/spool/mail/`
  - Remote access via pop3 or imap

16-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The mail delivery process begins when the user decides to send a composed message. The user's mail agent passes the message along to its configured MTA, usually a central mail gateway. With Sendmail 8.12, the user program calls **sendmail** as a non-privileged *mail submission program (MSP)* which will relay it to the MTA. This gateway reads the message and extracts the destination addresses from it. The MTA will unravel each email address, gathering networks, machines, and users to whom to send the message.

Once the MTA has verified all destination email addresses, it will notify the MUA that the mail was sent. Next, the MTA will deliver the message to the configured *mail exchanger (MX)* for each domain; should the primary one be down, the next MX for the domain will be chosen. If no mail exchangers are available (e.g., they're all down), then the MTA will queue the message and attempt delivery later.

When the message reaches the final destination, the target MTA hands the message to the system MDA. Under many systems, the MTA and the MDA are the same program, **sendmail**. The MDA will store the message in a spool file, or pass it through filters, or any perform whatever other instructions it is given for the particular site.

Users may then retrieve their mail either locally by reading from a spool file, or remotely, by using a protocol such as POP or IMAP.

# Simple Mail Transport Protocol

- RFC-standard protocol for talking to MTA's
  - Almost always uses TCP port 25
  - Extended SMTP (ESMTP) provides enhanced features for MTA's
  - An MTA often uses Local Mail Transport Protocol (LMTP) to talk to itself
- Example MSP:  

```
mail -vs 'Some Subject' student@stationX.example.com
```
- Use **telnet** to troubleshoot SMTP connections

16-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

You can simulate an SMTP delivery manually with **telnet**. Say you wanted to send email to bob@example.com:

```
[student@stationX]$ dig -t mx example.com | grep "MX"
example.com.      1D   IN   MX    0 lava.example.com.
```

```
[student@stationX]$ telnet lava.example.com 25
Trying 199.44.172.1...
Connected to lava.example.com.
Escape character is '^]'
220 example.com ESMTP Sendmail 8.11.6/8.11.6; Wed, 17 Oct
2006 12:31:04 -0400
HELO mynet.com
250 mynet.com Hello user@mynet.com [199.32.75.1], pleased
to meet you
MAIL From: user@mynet.com
250 user@mynet.com... Sender ok
RCPT To: bob@example.com
250 bob@example.com... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
test
.
```

```
250 RAA21966 Message accepted for delivery
```

```
QUIT
```

```
221 example.com closing connection
```

Note: SMTP commands (helo, mail from, rcpt to, and the others) ignore case by RFC standard.

LMTP is a simplified protocol that an MTA can use to deliver mail to recipients on localhost. This distinction will become important when implementing configurations since some settings apply *only* to SMTP and *not* to LMTP.

# Using alternatives to Switch MTAs

- Overview of the alternatives system
  - displays or configures the preferred MTA and associated man pages based on a *generic name*
  - *generic name* is a link to a link in `/etc/alternatives/`
  - only the links in `/etc/alternatives/` are modified
- Switching between MTA's
  - Stop the current MTA and disable boot-time startup
  - **alternatives --config mta** and make a selection
  - Start the new MTA and enable boot-time startup
- Graphical interface: **system-switch-mail-gnome** package

16-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 9700.

The **alternatives** system manages many packages in the distribution that provide the same service. For instance, both Sendmail and Postfix provide electronic mail support, but only one of them should be used at a time on a single machine.

With **alternatives**, an executable with a generic name on the filesystem is used to access a particular service. This executable is really a symbolic link to another symlink in the `/etc/alternatives/` directory. For example, `/usr/sbin/sendmail` is really a link to `/etc/alternatives/mta`. In order to select between Sendmail or Postfix, we just change the symlink for `/etc/alternatives/mta`. This is normally done with the **alternatives** command. Some example commands are listed below.

To display which MTA alternative is in use:

**alternatives --display mta**

To interactively choose from the available MTA alternatives from the command line:

**alternatives --config mta**

To script Postfix as the default mail system:

**alternatives --set mta /usr/sbin/sendmail.postfix**

You may use the GUI tool **system-switch-mail**, if installed, to make these changes. Note that it calls **alternatives** to effect the configuration, and it stops the old service, then starts the new service. This application is part of the **system-switch-mail** and **system-switch-mail-gnome** RPM.

# Mail Security

- MTAs listen on 25/tcp
- MDAs listen all or some of 110,143, 993, 995/tcp
- Only **/usr/sbin/sendmail** uses **tcp\_wrappers**
- Confined by SELinux as `sendmail_t`, `postfix_master_t`, `dovecot_t`, respectfully

16-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 625 2994 or +1 (919) 754 3700.

The MTA's operate on port 25/tcp. This port is configurable, but there is little use, as no other servers would be able to find the new port value to send mail. The port configuration for

```
[root@stationX ~]# nmap 192.168.0.X
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-02-01 01:02 EST
Interesting ports on www2.example.com (192.168.0.X):
Not shown: 1678 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
110/tcp   open  pop3
143/tcp   open  imap
993/tcp   open  imaps
995/tcp   open  pop3s
Nmap finished: 1 IP address (1 host up) scanned in 0.121 seconds
```

Sendmail runs as `/usr/sbin/sendmail`, and uses `libwrap.so`.

```
[root@stationX ~]# ldd /usr/sbin/sendmail | grep libwrap.so
libwrap.so.0 => /usr/lib64/libwrap.so.0 (0x00002aaaab5ab000)
```

Postfix runs as `/usr/libexec/postfix/master`, and spawns several child processes. Neither the parent, nor children use `libwrap.so`.

```
[root@stationX ~]# ldd /usr/libexec/postfix/master | grep libwrap.so
[root@stationX ~]# strings /usr/libexec/postfix/master | grep "hosts\..*"

```

Dovecot runs as `/usr/sbin/dovecot`, and spawns several child processes. Neither the parent, nor children use `libwrap.so`.

```
[root@stationX ~]# ldd /usr/sbin/dovecot | grep libwrap.so
[root@stationX ~]# strings /usr/sbin/dovecot | grep "hosts\..*"

```

Each of the processes are confined by SELinux, but have access to the same security contexts, as they share most of the same data.

- `etc_mail_t` = shared configuration files

- postfix\_etc\_t = postfix specific
- dovecot\_etc\_t = postfix specific
- mail\_spool\_t = mail messages
- var\_log\_t = log files

Postfix and Dovecot have an SELinux boolean:

- postfix\_disable\_trans = disable SELinux restriction of Postfix
- dovecot\_disable\_trans = disable SELinux restriction of Postfix

Sendmail is not allowed to run unconfined.



# Sendmail Configuration Files

- Configuration stored in `/etc/mail/`
- `sendmail.cf` is the main configuration file for Sendmail:
  - Contains domain alias directives, header rewriting directives, relaying rules, etc.
- `sendmail.mc` is a macro based configuration file for `sendmail.cf`
  - Processed using the **m4** command
  - Requires the `sendmail-cf` RPM
- `local-host-names`

16-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

`sendmail.cf` contains complicated configuration specifications for forwarding rule sets and mailer table references, and as a general rule should not be manually edited. Red Hat recommends, and uses, the **m4** configuration method described later in this unit.

`submit.cf` is an alternative, simpler version of the configuration file used only when **sendmail** is called as a MSP by a user program. It also should not be manually edited.

To install the **m4** macro compiler and the base **sendmail** m4 libraries, install the `m4` and `sendmail-cf` RPM packages.

A few important macros for `sendmail.mc`:

`DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')` - restricts sendmail to only listen for network connections on the loopback interface.

`FEATURE(`accept_unresolvable_domains')` - Accepts mail from unresolvable domains. Should be disabled on public MTAs.

`MAIL_HUB, SMART_HOST` - relaying configuration.

Sendmail also references `/etc/mail/access` for relay permissions.

# Incoming Sendmail Configuration

- Modify `/etc/mail/sendmail.mc` to listen on all interfaces

```
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```
- Add to `/etc/mail/local-host-names` each hostname by which the server may be referred
- Modify access control
  - Update `/etc/hosts.{allow,deny}`
  - Add an iptables rule to allow SMTP traffic
- Restart sendmail

16-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Sendmail will listen on all interfaces if a specific interface is not declared, so comment out the default specification for the loopback interface, restart sendmail, and double-check:

```
[root@stationX ~]# netstat -tulpn | grep sendmail
tcp    0    0 0.0.0.0:25      0.0.0.0:* LISTEN      19938/sendmail
```

In order to determine if **sendmail** is identifying your station's hostname correctly, invoke it with its debugging command line switch set to 0:

```
[root@stationX]# sendmail -d0 < /dev/null
Version 8.13.1
Compiled with: DNSMAP HESIOD HES_GETMAILHOST LDAPMAP LOG
MAP_REGEX MATCHGECOS MILTER MIME7TO8 MIME8TO7 NAMED_BIND
NETINET NETINET6 NETUNIX NEWDB NIS PIPELINING SASLv2 SCANF
STARTTLS TCPWRAPPERS USERDB USE_LDAP_INIT

===== SYSTEM IDENTITY (after readcf) =====
  (short domain name) $w = stationX
 (canonical domain name) $j = stationX.example.com
   (subdomain name) $m = example.com
      (node name) $k = stationX.example.com
=====
```

Recipient names must be specified

If sendmail is returning your station name as `localhost`, you probably have a misconfigured `/etc/hosts`. Examine your `/etc/hosts`, and remove all but the `localhost` hostname references, and try again. If `/etc/hosts` appears correct, check the definition of `HOSTNAME` in `/etc/sysconfig/network`.

# Sendmail Operation

- `/etc/mail/local-host-names`
  - must contain server's name and aliases
- **mail -v user**
  - view SMTP exchange with local relay
- **mailq and mailq -Ac**
  - view messages queued for future delivery
- **sendmail -q**
  - reprocess the email queue
- **tail -f /var/log/maillog**
  - View log in real-time

16-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Sendmail will not accept mail for local delivery for hosts that are not specified by name in `/etc/mail/local-host-names`. This file contains a list of host names which, if seen in an e-mail address, should be delivered locally. The host names should resolve to your server or have a MX record pointing to your server.

In Sendmail 8.12, the **mail -v** command displays the SMTP exchange between the MSP and the local relay MTA only. It is useful to debug local **sendmail** daemon configuration. To send a test message and view the SMTP exchange, use

```
mail -v user
```

type the message and press *Ctrl-d* to send. (In older versions of Sendmail, there was no unprivileged MSP -- the user's e-mail program ran a set-uid MTA directly, and **mail -v** instead displayed the local MTA to remote MTA communication.)

The **mailq** command is useful for displaying messages waiting in a queue for delivery. By default, it displays the queue of messages waiting to be processed by your local MTA for delivery or relay to a remote MTA. Messages can sit in this queue for a long time if the remote host is refusing connections. If you run **mailq -Ac**, the queue of messages waiting to be sent by your MSP to the local MTA relay will be displayed instead. Messages can sit in this queue for a long time if your local host is having problems with name resolution.

To reprocess the mail queue, run the **sendmail -q** command.

During any server configuration or testing, monitoring the appropriate logs with **tail -f** can be invaluable. Note that the default `/etc/syslog.conf` file precedes the `/var/log/maillog` target with a '-' (dash) symbol to disable syncing the file after every logging. You can remove the dash to enable syncing and get real-time logging from the **tail -f** command.

# Incoming Postfix Configuration

- Modify `/etc/postfix/main.cf`

- Listen on all interfaces

```
inet_interfaces = all
```

- Specify each name and alias by which the server may be referred

```
mydestination = $myhostname, localhost.$mydomain,  
                localhost, $mydomain
```

- Add iptables rules to allow SMTP traffic
- Restart postfix

16-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Since key/value pairs are evaluated in order (last one wins):

1. Uncomment the line `inet_interfaces = all`
2. Add a comment character (#) in front of the subsequent line `inet_interfaces = localhost`
3. **service postfix restart**
4. Confirm that Postfix is listening on all interfaces:

```
[root@stationX ~]# netstat -tulpn | grep master  
tcp      0      0 0.0.0.0:25          0.0.0.0:* LISTEN      16179/master
```

For examples of firewall rules for SMTP, see the previous slides on SMTP and Incoming Sendmail Configuration.

For testing proper MTA operation, always test from a remote machine. This provides several benefits:

- Tests intervening firewalls for end-to-end connectivity
- Tests application-layer access controls
- Ensures that SMTP is being used (as opposed to LMTP)

# Postfix Operation

- `main.cf` settings
  - Server names: `mydestination` must contain server's name and aliases
  - Listening interfaces: `inet_interfaces = all`
  - Archive all messages: `always_bcc = address`
- View SMTP exchange: `mail -v user@domain.tld`
- View deferred messages: `postqueue -p`
- Flush deferred messages: `postqueue -f`
- Follow log: `tail -f /var/log/maillog`

16-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Postfix will not accept mail for local delivery for hosts that are not specified by name on the `mydestination` line.

```
mail -v user
```

type the message and press *Ctrl-d* to send.

Run either `postqueue -f` or `postsuper -r ALL` to reprocess (flush) the mail queue.

During any server configuration or testing, monitoring the appropriate logs with `tail -f` can be invaluable. Note that the default `/etc/syslog.conf` file precedes the `/var/log/maillog` target with a '-' (dash) symbol to disable syncing the file after every logging. You can remove the dash to enable syncing and get real-time logging from the `tail -f` command.

The `always_bcc` option may provide an easy way for organizations to comply with requirements to archive all electronic messages both in and out of the company.

# Email Aliases

- Aliases allow mail for one local user name to deliver to another
- Defined in `/etc/aliases`
- After updating aliases, run **newaliases**

16-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 826 2994 or +1 (919) 754 3700.

`/etc/aliases` defines a list of local user aliases. Should user `marc` need `root`'s email, `/etc/aliases` would contain the following line:

```
root: marc
```

Every time any sendmail or postfix database like `/etc/aliases` is changed, it must be rehashed to a database format. For `/etc/aliases` this is done with **newaliases**. For other sendmail database files this is done by running **make** in `/etc/mail/`. For postfix, you can use the **postmap** command. These database files are also rehashed as needed whenever the daemons are started or restarted using their initialization scripts or the **service** command.

# Mail Retrieval Protocols

- Post Office Protocol
  - All data, including passwords, is passed in clear text over TCP port 110
  - Use POP3s to provide SSL encryption of data over TCP port 995
- Internet Mail Access Protocol
  - All data, including passwords, is passed in clear text over TCP port 143
  - Use IMAPs to provide SSL encryption of data over TCP port 993
- Dovecot supports POP3, POP3s, IMAP, and IMAPs

16-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

POP and IMAP are RFC-standard protocols for mail retrieval. IMAP provides a number of significant improvements over POP3. A few are provided here:

- Tagged commands for network pipelines
- Multiple folders
- Additional login mechanisms

Use SSL to protect passwords and data when using POP or IMAP. Here's a network trace captured when SSL was *not* enabled:

```
[root@stationX ~]# tshark -ni eth0 -R "tcp.srcport eq 110 or tcp.dstport eq 110"
53.923996 192.168.0.198 -> 192.168.0.254 POP Response: +OK Dovecot ready.
53.924218 192.168.0.254 -> 192.168.0.198 TCP 33467 > 110 [ACK] Seq=1 ...
57.049295 192.168.0.254 -> 192.168.0.198 POP Request: user test
57.049407 192.168.0.198 -> 192.168.0.254 TCP 110 > 33467 [ACK] Seq=21 ...
57.050050 192.168.0.198 -> 192.168.0.254 POP Response: +OK
57.050262 192.168.0.254 -> 192.168.0.198 TCP 33467 > 110 [ACK] Seq=12 ...
64.113221 192.168.0.254 -> 192.168.0.198 POP Request: pass mypassword
64.153092 192.168.0.198 -> 192.168.0.254 TCP 110 > 33467 [ACK] Seq=26 ...
64.180158 192.168.0.198 -> 192.168.0.254 POP Response: +OK Logged in.
```

# Dovecot Configuration

- Listens on all IPv6 and IPv4 interfaces by default
- Specify protocols in `/etc/dovecot.conf`
  - `protocols = imap imaps pop3 pop3s`
- Make a private key and self-signed certificate before using SSL
  1. Confirm system time to avoid date issues
  2. Review `/etc/dovecot.conf` for key and certificate locations
  3. Run **make -C /etc/pki/tls/certs dovecot.pem**
    - Creates a single PEM file containing both the key and the certificate
  4. Copy the new PEM file to both locations

16-14

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Confirm the listening ports for Dovecot:

```
[root@stationX ~]# netstat -tulpn | grep dovecot
tcp        0      0 0.0.0.0:993 0.0.0.0:* LISTEN  9912/dovecot
tcp        0      0 0.0.0.0:995 0.0.0.0:* LISTEN  9912/dovecot
tcp        0      0 0.0.0.0:110 0.0.0.0:* LISTEN  9912/dovecot
tcp        0      0 0.0.0.0:143 0.0.0.0:* LISTEN  9912/dovecot
```

## SSL Background

Public key infrastructure (PKI) systems are based on asymmetric algorithms and mathematically-related keys such that data encrypted by a private key can only be decrypted by the corresponding public key. Similarly, data encrypted with the public key can only be decrypted using the private key.

Although the asymmetric algorithms are computationally intensive---and therefore slow---PKI can be used to create and distribute a random, shared secret key in order to use faster, symmetric algorithms between two hosts. This random key is typically referred to as a session key and is used only temporarily.

As seen in `/etc/pki/tls/certs/Makefile`, running **make dovecot.pem** creates a private key, then uses that key to create a self-signed certificate (i.e., subject and issuer of certificate are the same, as seen on the next page). The certificate is essentially the public key. The make process then combines the two elements, private and public keys, into a single ASCII file using *Privacy Enhanced Mail* (PEM) format.

In this course it will be sufficient to use a self-signed certificate for the Dovecot SSL configuration. The self-signed certificate demonstrated here can also be used securely for an in-house trusted environment, but you would want to separate the key from the certificate. The key must be known only by dovecot. The self-signed certificate can be distributed to internal hosts as a trusted CA root certificate for certificate verification.



For externally-facing SSL configurations, one would prefer to use a digital certificate signed by a trusted certificate authority since external clients will already have the CA root certificates for those trusted authorities.

# Verifying IMAP Operation

- Verifying server operation
  - Graphical: Thunderbird and Evolution
  - Text-mode: Mutt and Fetchmail

```
mutt -f imap://user@server[:port]
mutt -f imaps://user@server[:port]
```

- Can also use **telnet** (IMAP) or **openssl s\_client** (IMAPs)
  - Identify problems with certificate date or permissions

16-15

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

```
[student@stationX ~]$ openssl s_client -connect stationX.example.com:993
...output truncated...
Certificate chain
 s:/C=US/ST=Texas/L=Austin/O=Example, Inc./CN=stationX.example.com...truncated
 i:/C=US/ST=Texas/L=Austin/O=Example, Inc./CN=stationX.example.com...truncated
...output truncated...
    Start Time: 1169831361
    Timeout    : 300 (sec)
    Verify return code: 9 (certificate is not yet valid)
---
+OK Dovecot ready.
```

This example demonstrates using **openssl s\_client** to recognize a problem with certificate dates. In this case, the `dovecot.pem` was created while the system clock was set wrong. Sometime after creating the PEM file, the admin fixed the system date. Using **mutt** to verify the certificate may not highlight the problem since **mutt** simply shows the certificate to the user and asks for confirmation, then continues normally.

When configuring SSL services, always:

- Confirm your system clock first
- Watch for typo's when creating certificates
- Test!

## End of Unit 16

- Questions and Answers
- Summary
  - system-switch-mail
  - sendmail, sendmail-cf, m4
  - postfix
  - /etc/aliases, newaliases
  - dovecot, openssl

16-16

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 16

## Mail Services

---

**Goal:** Understand the function, deployment, and configuration on the mail services components.

**System Setup:** Throughout this lab, the hostnames and domain names that you use will be based on the IP address of your machine. Any time the lab refers to stationX, you should replace X with your station number. Any references to stationY will mean an unprivileged account on a classmate's system, or a remote station identified by your instructor.

Your `tcp_wrappers` configuration should include the following:

```
# cat /etc/hosts.allow
ALL:127.0.0.1 [:::1]
sshd:192.168.0. [fe80::]/64
```

Your `iptables` firewall rules should include the following:

```
# cat /etc/sysconfig/iptables
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/255.255.255.0 -p tcp -m tcp
    --dport 22 -j ACCEPT
-A INPUT -p udp -m udp --sport 53 -j ACCEPT
-A INPUT -s 192.168.0.254 -j ACCEPT
-A INPUT -j LOG
-A INPUT -j REJECT --reject-with icmp-port-unreachable
```

Your `/etc/resolv.conf` should be set as follows:

```
search example.com
nameserver 192.168.0.254
```

Test as follows:

```
# dig +short stationX.example.com
192.168.0.X
```

## Sequence 1: Configure Sendmail as an MTA

**Scenario:** A site needs a local e-mail server to support their sub-domain of stationX.example.com. Implement security as appropriate.

**Deliverable:** An Sendmail server for stationX.example.com.

**Instructions:**

1. Sendmail is operational as part of a base install, so no installation is needed. It is, however, configured to only accept local connections. For troubleshooting purposes, launch a log monitoring window:

```
# xterm -e "tail -f /var/log/maillog" &
```

Start with **iptables** stopped. Use **telnet** to connect to localhost 25. Once connected, type quit to disconnect. Now, use **telnet** to connect to 192.168.0.X 25. This should fail.

2. To see why the connection was refused, run **nmap** against both localhost and 192.168.0.X 25. Notice that port 25 is only open for local connection. To open sendmail for remote connection, install **sendmail-cf** and comment the loopback restriction in **sendmail.mc**. Restart sendmail and execute **nmap 192.168.0.X** to verify the ports are open.
3. The server is now working locally. Start **iptables** and connect to stationY. From stationY, execute **nmap 192.168.0.X -p25**. This should indicate the port is not open. Check /var/log/messages for clues. Correct the problem.
4. From stationY, execute **telnet 192.168.0.X 25**. Everything should seem correct, until you try a command (**help**, for instance) in the session. This should fail. Review the log file, determine the problem, and fix it.
5. Time to test a real message from stationY:

```
$ echo | mail -s "Sendmail: $HOSTNAME" root@stationX.example.com
```

On stationX, run **mutt** as root to view the message.

## Sequence 2: Migrating to Postfix

Scenario: A facility using Sendmail has decided to migrate to Postfix as their e-mail server. Install and implement Postfix, with appropriate security.

Deliverable: A Postfix server for `stationX.example.com`.

### Instructions:

1. Starting with **iptables** stopped, install Postfix. Also installing **system-switch-mail** will greatly simplify the migration.
2. Execute **system-switch-mail**, select Postfix, and commit the changes. Verify Sendmail's current status and boot status. Verify Postfix's current status and boot status.
3. As with Sendmail, test Postfix by using **telnet** to connect to `localhost 25`. Next, connect to `192.168.0.X 25`. This should fail. The reason this failed is because **system-switch-mail** does not migrate any configuration settings, it only modifies the daemon control mechanisms.
4. With Sendmail, it was necessary to remove the loopback restriction. The same is true with Postfix. In `/etc/postfix/main.cf`, locate the *inet\_interfaces* section. Comment out the `localhost` restriction, and uncomment the `all` line. Restart the service and test with **telnet**. This time it should connect.
5. The server is now working locally. Start **iptables** and connect to `stationY`. Execute **nmap 192.168.0.X -p25**. It should report "open". Why isn't **iptables** blocking the connection to Postfix? What about **tcp\_wrappers**?
6. Time to test a real message from `stationY`:  

```
$ echo | mail -s "Postfix: $HOSTNAME" root@stationX.example.com
```

  
On `stationX`, run **mutt** as root to view the message.

## Sequence 3: Adding new aliases

**Scenario:** Your organization needs a series of generic e-mail accounts that will redirect to actual users. Use mail aliases to implement this feature.

**Deliverable:** A mail aliases distribution system.

**Instructions:**

1. Verify you can login to the "student" account. Add the following lines to the `/etc/aliases` file:

```
me: student
wizards: root, me
methere: student@stationX.example.com
```

Run the **newaliases** command to update the alias database.

2. Send a mail message to the recipient aliases that you defined.

```
# newaliases
# echo "hello" | mail -s "hello, me" me
# echo "hello" | mail -s "hello, wizards" wizards
# echo "hello" | mail -s "hello, methere" methere
```

3. Switch to the student account and use mutt to view student's mail. You should see three messages. Return to the root account and use mutt to view root's mail. You should see the wizards message.

## Sequence 4: Installing the Dovecot MDA.

**Scenario:** Remote users need to access mail on the local server. Configure Dovecot to allow outside connection on all protocols. The encrypted protocols should use the default certificate.

**Deliverable:** A functioning MDA server.

### Instructions:

1. Starting with iptables down, install **dovecot**, ensure it is running, and configured to start at boot time.
2. We know that dovecot is a multi-protocol MDA. Use **nmap** to scan your systems open ports. To test the access to the mail spool, use the following:  
  

```
mutt -f pop://student@localhost
```
3. At this point, everything is working. The problem we face, however, is that the username and password need to retrieve the mail, was sent across the network in clear text. To verify this, install **wireshark**, and "sniff" the authentication sequence.
4. To protect the password, disable the insecure protocols within **dovecot**. Restart the service. Close the original tshark monitoring window and launch a new sniff of the pop3s port. Authenticate again and watch the sniff.
5. At this point the server is working locally. Close the monitoring window, start **iptables**, login to stationY, and attempt to connect. This should fail. Find and correct the problem.



## Sequence 5: Creating a unique Dovecot certificate

Scenario: Concerned about the ability to sniff passwords from insecure protocols, it has been decided to use a site specific encryption certificate.

Deliverable: A secure MDA server.

### Instructions:

1. At install, the Dovecot RPM generates a generic set of encryption keys. Since they are generic, they are susceptible to "man in the middle attacks". It might be more appropriate to generate a site specific set of keys. On Red Hat Enterprise Linux 5, the keys are stored in the `/etc/pki` directory. Move to the directory and execute `ls -l dovecot/*` to view the keys.

2. List the `tls/certs/` directory. You will see a `Makefile` that can be used to generate unique keys. Execute:

```
make -C tls/certs/ dovecot.pem
```

At the prompts, use your current location information, and use "example.com" for the organizational information. Use `stationX.example.com` as your station name and `root@stationX.example.com` as your email address.

3. Again, list the `tls/certs/` directory. Notice there is a new `dovecot.pem` file. This file needs to replace the two generic `dovecot.pem` files located earlier.
4. Restart Dovecot to force it to read the keys. Use **mutt** to connect to the POP3S server. This time the certificate should display your information rather than the generic information.

## Sequence 1 Solutions

1. Sendmail is operational as part of a base install, so no installation is needed. It is, however, configured to only accept local connections. Use **telnet** to connect to localhost 25. Once connected, type quit to disconnect.

```
# telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to stationX (127.0.0.1).
Escape character is '^]'.
220 localhost.localdomain ESMTP Sendmail 8.13.8/8.13.8; Mon, 26
  Feb 2007 10:36:31 -0500
quit
221 2.0.0 localhost.localdomain closing connection
Connection closed by foreign host.
```

Now, use **telnet** to connect to 192.168.0.X 25. This should fail.

```
# telnet 192.168.0.X 25
Trying 192.168.0.X...
telnet: connect to address 192.168.0.X: Connection refused
telnet: Unable to connect to remote host: Connection refused
```

2. To see why the connection was refused, run **nmap** against both localhost and 192.168.0.X 25.

```
# nmap localhost
... output truncated ...
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
# nmap 192.168.0.19
... output truncated ...
PORT      STATE SERVICE
22/tcp    open  ssh
```

Notice that port 25 is only open for local connection. To open sendmail for remote connection, install **sendmail-cf** and comment the loopback restriction in **sendmail.mc**.

```
# yum install -y sendmail-cf
... output truncated ...
Installed: sendmail-cf 0:8.13.8-2.el5
Complete!
# cd /etc/mail
# grep 127. sendmail.mc
dn1 # 127.0.0.1 and not on any other network devices. Remove the
  loopback
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dn1
# vi sendmail.mc
```

Located the loopback restriction line, and comment it out by placing DNL # at the beginning.

```
# grep 127. sendmail.mc
dnl # 127.0.0.1 and not on any other network devices. Remove the ✓
loopback
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

Restart sendmail and execute **nmap 192.168.0.X** to verify the ports are open.

```
# nmap 192.168.0.19
... output truncated ...
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
```

3. The server is now working locally. Start iptables and connect to stationY. From stationY, execute **nmap 192.168.0.X -p25**. (The extra option of -p25 checks only the specified port, yielding a much faster response.)

```
[stationY]$ nmap 192.168.0.X -p25
... output truncated ...
PORT      STATE SERVICE
25/tcp    closed smtp
```

Since we know that application works locally **tail /var/log/messages** and look for log entries from **iptables**. There should be an entry similar to the following:

```
Feb 28 22:06:08 stationX kernel: IN=eth0 OUT= ✓
MAC=00:16:3e:03:10:03:00:0d:60:fa:f5:f2:08:00 SRC=192.168.0.Y ✓
DST=192.168.0.X LEN=44 TOS=0x00 PREC=0x00 TTL=51 ID=24117 ✓
PROTO=TCP SPT=43243 DPT=25 WINDOW=4096 RES=0x00 SYN URGP=0
```

Note the TCP and DPT fields, specifying the protocol and destination ports, respectively. Construct an **iptables** rule to open the firewall. It may be similar to the following:

```
# iptables -I INPUT 5 -p tcp --dport 25 -j ACCEPT
# service iptables save
[stationY]$ nmap 192.168.0.X -p25
... output truncated ...
PORT      STATE SERVICE
25/tcp    open  smtp
```

Because of the "open nature" of e-mail, this example is probably reasonable for our server. Should we need to exclude specific servers, DROP rules could be add *before* the ACCEPT rule.

4. From stationY, execute **telnet 192.168.0.X 25**. Everything should seem correct, until you try a command (**help**, for instance) in the session. This should fail.

```
# telnet 192.168.0.19 25
```

```
... output truncated ...
220 stationX.example.com ESMTP Sendmail 8.13.8/8.13.8; Wed, 28 ✓
Feb 2007 17:37:54 -0500
help
550 5.0.0 Access denied
```

Review the mail log file. There should be an entry similar to the following:

```
Feb 28 17:37:54 stationX sendmail[1813]: l1SMbs8I001813: ✓
tcpwrappers (stationX.example.com, 192.168.0.19) rejection
```

Notice that sendmail reported a **tcp\_wrappers** rejection. Add an entry to `/etc/hosts.allow` for sendmail. Since this line will establish who can send mail, it will usually be very open. We could, however use it to restrict certain domains:

```
sendmail : ALL except .aq
```

This would accept all traffic except that coming from Antarctica. (Yes, Antarctica really has it's own domain.) Try **telnet 192.168.0.X 25** again.

```
# telnet 192.168.0.X 25
... output truncated ...
helo stationX
250 stationX.example.com Hello stationY.example.com ✓
[192.168.0.Y], pleased to meet you
quit
```

5. Time to test a real message from stationY:

```
[stationY]$ echo | mail -s "Sendmail: $HOSTNAME" ✓
root@stationX.example.com
```

On stationX, run **mutt** as root to view the message.

```
[stationY]# mutt
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
1 N Feb 26 root ( 0) Sendmail: stationY.example.com
```

## Sequence 2 Solutions

1. Starting with **iptables** stopped, install Postfix. Also installing **system-switch-mail** will greatly simplify the migration.

```
# service iptables stop
Flushing firewall rules:           [ OK ]
Setting chains to policy ACCEPT: filter [ OK ]
Unloading iptables modules:       [ OK ]
[root@stationX mail]# yum install -y postfix system-switch-mail
... output truncated ...
Installed: postfix 2:2.3.3-2 system-switch-mail.noarch ✓
0:0.5.25-12
Complete!
```

2. Execute **system-switch-mail**.

```
# system-switch-mail
```

Select Postfix, and commit the changes. Verify Sendmail's current status and boot status.

```
# service sendmail status
sendmail is stopped
# chkconfig --list sendmail
service sendmail supports chkconfig, but is not referenced in
any runlevel (run 'chkconfig --add sendmail')
```

Verify Postfix's current status and boot status.

```
# service postfix status
master (pid 2855) is running...
# chkconfig --list postfix
postfix    0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

3. As with Sendmail, test Postfix by using **telnet** to connect to localhost 25.

```
# telnet 127.0.0.1 25
... output truncated ...
220 stationX.localdomain ESMTP Postfix
quit
```

Next, connect to 192.168.0.X 25. This should fail.

```
# telnet 192.168.0.X 25
Trying 192.168.0.X...
telnet: connect to address 192.168.0.X: Connection refused
telnet: Unable to connect to remote host: Connection refused
```

The reason this failed is because **system-switch-mail** does not migrate any configuration settings, it only modifies the daemon control mechanisms

4. With Sendmail, it was necessary to remove the loopback restriction. The same is true with Postfix. In `/etc/postfix/main.cf`, locate the `inet_interfaces` section. Comment out the `localhost` restriction, and uncomment the `all` line.

```
# grep "^inet_interfaces" /etc/postfix/main.cf
inet_interfaces = localhost
# vi /etc/postfix/main.cf
# grep "^inet_interfaces" /etc/postfix/main.cf
inet_interfaces = all
```

Restart the service and test with **telnet**. This time it should connect.

```
# service postfix restart
Shutting down postfix:           [ OK ]
Starting postfix:                 [ OK ]
# telnet 192.168.0.X 25
... output truncated ...
220 stationX.localdomain ESMTP Postfix
quit
```

5. The server is now working locally. Start **iptables**, connect to `stationY` and execute **nmap 192.168.0.X -p25**.

```
# service iptables start
Applying iptables firewall rules: [ OK ]
[stationY]$ nmap 192.168.0.19 -p25
... output truncated ...
PORT      STATE SERVICE
25/tcp    open  smtp
```

It should report "open". Why isn't **iptables** blocking the connection to Postfix? What about **tcp\_wrappers**?

Recall that in the previous exercise, you build a rule to allow TCP 25. (You did save those rules, didn't you?) Port 25 is SMTP, not Sendmail. Since Postfix is also an SMTP server, it uses the same rule. Since Postfix does not use `libwrap`, there is no **tcp\_wrappers** protection.

6. Time to test a real message from `stationY`:

```
$ echo | mail -s "Postfix: $HOSTNAME" root@stationX.example.com
```

On `stationX`, run **mutt** as root to view the message.

```
[stationY]# mutt
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
  1 N   Feb 26 root      (  0) Sendmail: stationY.example.com
  1 N   Feb 26 root      (  0) Postfix: stationY.example.com
```

## Sequence 3 Solutions

1. Verify you can login to the "student" account. Add the following lines to the `/etc/aliases` file:

```
me: student
wizards: root, me
methere: student@stationX.example.com
```

Run the **newaliases** command to update the alias database.

```
# newaliases
```

2. Send a mail message to the recipient aliases that you defined.

```
# newaliases
# echo "hello" | mail -s "hello, me" me
# echo "hello" | mail -s "hello, wizards" wizards
# echo "hello" | mail -s "hello, methere" methere
```

3. Switch to the student account and use mutt to view student's mail. You should see three messages.

```
# su - student
$ mutt
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group
  1 N   Mar 13 root      (  1) hello, me
  2 N   Mar 13 root      (  1) hello, wizards
  3 N   Mar 13 root      (  1) hello, methere
$ exit
```

Return to the root account and use mutt to view root's mail. You should see the wizards message.

```
# mutt
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group
  1 N F Mar 13 To wizards@stat (  1) hello, wizards
```

## Sequence 4 Solutions

1. Starting with iptables down, install **dovecot**, ensure it is running, and configured to start at boot time.

```
# service iptables stop
... output truncated ...
# yum install -y dovecot
... output truncated ...
Installed: dovecot 0:1.0-1.2.rc15.el5
Dependency Installed: mysql 0:5.0.22-2.1 perl-DBI 0:1.52-1.fc6
    postgresql-libs 0:8.1.4-1.1
Complete!
# service dovecot status
dovecot is stopped
# chkconfig dovecot --list
dovecot    0:off  1:off  2:off  3:off  4:off  5:off  6:off
# service dovecot start
Starting Dovecot Imap:
# chkconfig dovecot on
```

2. We know that dovecot is a multi-protocol MDA. Use **nmap** to scan your systems open ports.

```
# nmap localhost
... output truncated ...
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
110/tcp   open  pop3
143/tcp   open  imap
993/tcp   open  imaps
995/tcp   open  pop3s
```

Notice that by default, dovecot has opened all four delivery protocols. Using **mutt** and POP, test access to the mail pool.

```
# mutt -f pop://student@localhost
Password for student@localhost:
connecting...
```

3. At this point, everything is working. The problem we face, however, is that the username and password need to retrieve the mail, was sent across the network in clear text. To verify this, install **wireshark**, and "sniff" the authentication sequence.

```
# yum install -y wireshark
... output truncated ...
Installed: wireshark 0:0.99.4-2.el5
Complete!
# xterm -e "tshark -i lo port pop3" &
```



This command will open **tshark** (Text Shark) in another window. It will monitor the localhost interface for traffic moving through the pop3 port. Login to the dovecot server again.

```
# mutt -f pop://student@localhost
Password for student@localhost:
connecting...
```

In the **tshark** there should be some line similar to the following:

```
... lines trimmed for brevity ...
POP Request: USER student
POP Response: +OK
TCP 42477 > pop3 [ACK] Seq=21 Ack=99 Win=32800 Len=0
TSV=3783655 TSER=3783655
POP Request: PASS student
POP Response: +OK Logged in.
```

Notice the line indicating PASS. Notice the clear text password.

4. To protect the password, disable the insecure protocols within **dovecot**. Restart the service.

```
# grep "protocols" /etc/dovecot.conf
#protocols = imap imaps pop3 pop3s
# vi /etc/dovecot.conf
# grep "protocols" /etc/dovecot.conf
protocols = imaps pop3s
# service dovecot restart
Stopping Dovecot Imap: [ OK ]
Starting Dovecot Imap: [ OK ]
```

Close the original tshark monitoring window and launch a new sniff of the pop3s port.

```
# xterm -e "tshark -i lo port pop3s" &
```

Authenticate again and watch the sniff.

```
# mutt -f pops://student@localhost
... output truncated ...
This certificate belongs to:
imap.example.com
Unknown
Unknown
IMAP server
Unknown
... output truncated ...
(r)ect, accept (o)nce
Password for student@localhost:
```

Notice that **tshark** can not display the password information, because the transaction has been encrypted

5. At this point the server is working locally. Close the monitoring window and start **iptables**.

```
# service iptables start
```

Login to stationY, and attempt to connect. This should fail.

```
[stationY]$ mutt -f pops://student@stationX.example.com
Error connecting to server: stationX.example.com
```

Return to stationX and **tail /var/log/messages**. You should see a line similar to the following:

```
Mar  2 17:53:10 stationX kernel: IN=eth0 OUT=  ↵
MAC=00:16:3e:03:10:03:00:50:bf:19:7a:07:08:00 SRC=192.168.0.254  ↵
DST=192.168.0.19 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=45675 DF  ↵
PROTO=TCP SPT=52254 DPT=995 WINDOW=5840 RES=0x00 SYN URGP=0
```

In previous exercises, we opened port 25 on the firewall, but have not opened ports for POP3S or IMAPS. Notice the **PROTO** and **DPT** information in the error message. Construct an **iptables** rule to allow POP3S to connect. While you're at it, make a rule for IMAPS, too. To find the port number, look at your earlier **nmap** scan.

```
# iptables -I INPUT 5 -p tcp --dport 995 -j ACCEPT
# iptables -I INPUT 6 -p tcp --dport 993 -j ACCEPT
# service iptables save
```

From stationY test the changes.

## Sequence 5 Solutions

1. At install, the Dovecot RPM generates a generic set of encryption keys. Since they are generic, they are susceptible to "man in the middle attacks". It might be more appropriate to generate a site specific set of keys. On Red Hat Enterprise Linux 5, the keys are stored in the `/etc/pki` directory. Move to the directory and execute `ls -l dovecot/*` to view the keys.

```
# cd /etc/pki
# ls
CA dovecot nssdb rpm-gpg tls
# ls -l dovecot/*
-rw-r--r-- 1 root root 496 Oct 13 14:25 \
dovecot/dovecot-openssl.cnf
dovecot/certs:
total 8
-rw----- 1 root root 847 Feb 26 13:44 dovecot.pem
dovecot/private:
total 8
-rw----- 1 root root 887 Feb 26 13:44 dovecot.pem
```

2. List the `tls/certs/` directory. You will see a Makefile that can be used to generate unique keys.

```
# ls tls/certs/
ca-bundle.crt make-dummy-cert Makefile
```

To start the process execute:

```
# make -C tls/certs/ dovecot.pem
```

At the prompts, use your current location information, and use "example.com" for the organizational information. Use `stationX.example.com` as your station name and `root@stationX.example.com` as your email address.

3. Again, list the `tls/certs/` directory. Notice there is a new `dovecot.pem` file.

```
# ls tls/certs/
ca-bundle.crt dovecot.pem make-dummy-cert Makefile
```

This file needs to replace the two generic `dovecot.pem` files located earlier.

```
# cp tls/certs/dovecot.pem dovecot/certs/dovecot.pem
cp: overwrite `dovecot/certs/dovecot.pem'? y
# cp tls/certs/dovecot.pem dovecot/private/dovecot.pem
cp: overwrite `dovecot/private/dovecot.pem'? y
```

4. Restart Dovecot to force it to read the keys. Use **mutt** to connect to the POP3S server. This time the certificate should display your information rather than the generic information.

```
# service dovecot restart
Stopping Dovecot Imap:
```

[ OK ]

Starting Dovecot Imap:

[ OK ]

# mutt -f pops://student@stationX.example.com

# Unit 17

## Troubleshooting

17-1

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Objectives

Upon completion of this unit, you should be able to:

- Develop a strategy for troubleshooting
- Fix problems in different areas of the Linux system
- Boot the system into various runlevels
- Use the Rescue environment

17-2

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Method of Fault Analysis

- Characterize the problem
- Reproduce the problem
- Find further information
- Eliminate possible causes
- Try the easy things first
- Backup config files before changing

17-3

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Begin by characterizing the problem. What are the symptoms? Is anything else "odd" about the system? Note suspicious log file entries and error messages.

Try to reproduce the problem. Does the problem still occur after the service is restarted? Rebooting the system is usually not necessary. If you suspect a problem in the start up sequence it is often enough to bring the system to single user mode and back up again.

Eliminate possible causes for the problem. For each possible reason for the error, find a way to confirm or refute the cause. Always start with the reason that is easiest to test.

Always backup the old configuration before changing it. This allows you to easily revert to the old config when a change does not solve the problem.

# Fault Analysis: Gathering Data

- Useful commands
  - **history**
  - **grep**
  - **diff**
  - **find /dir-cmin -60**
  - **strace command**
  - **tail -f logfile**
- Generate additional information
  - \*.debug in syslog
  - --debug option in application

17-4

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

**history** displays the last 1000 commands of a user. With **grep** you can find strings in text files.

**diff** shows the difference of two text files. Thus you can easily see what directives have been changed since the last backup of the file.

**find start-dir-cmin -60** lists all files that have been modified within the last 60 minutes.

**strace** can be used to gather further data from applications. The syntax is **strace command** when running the command, or **strace -p PID** when connecting to a running process.

**tail -f** can be used to follow log file as they are generated. This allows you to see the errors as they are being generated, which may shed some light on what the problem may be.

**syslog** can capture debug information from the kernel and from system services. The default setting in `/etc/syslog.conf` is to send all message of info or higher to `/var/log/messages`. To get all messages (debug and higher), add this line to the configuration file and restart **syslog**

```
/etc/syslog.conf:  
*.debug /var/log/debug
```

You may also want to add an file to `/etc/logrotate.d/` to rotate that file.

Most applications have a debugging option (e.g., `--debug`, `-d`, etc.) that forces the program to run in debug mode, thus creating copious entries in your log files. These options are often configured using the files in `/etc/sysconfig`.

For instance, to run **xinetd** in debug mode, add this entry to `/etc/sysconfig/xinetd`, and restart the service:



EXTRAOPTIONS="-d"

[root@stationX]# **service xinetd restart**

## Things to Check: X

- Never debug X while in runlevel 5!
- When changing hardware, try **system-config-display** first
- **X -probeonly**
- Is `/home` or `/tmp` full, or has the user reached a hard quota?
- Is **xfs** running?

17-5

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <ctraining@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

X.org has become quite good at hardware auto-detection. If you do need to reconfigure X, **system-config-display** provides a good starting point, and can be run either from the console or from within X itself. If you anticipate making multiple changes, you may find it more convenient to work on X from runlevel 3, using **startx** to test your changes.

Sometimes viewing the output of the X command itself is revealing. The **probeonly** switch will perform all tasks necessary to start the X server without actually starting it. Thus, the start up messages are displayed. If the text scrolls off the screen, the material off screen can be viewed by holding down the *Shift* key and pressing the *Page Up* key (*Shift-PgUp* to scroll down again). The output may be viewed by holding down *Shift* and using *Page Up* and *Page Down*. `/usr/share/hwdata/Cards` may provide useful information about optional configuration features for your hardware.

A user's inability to create files in the user's home directory or in `/tmp` either because of a full filesystem or because of a hard quota limit typically inhibits the ability of the user to run X. Although the exact symptoms differ between runlevels, messages will appear stating (in runlevel 3) or suggesting (in runlevel 5) that a filesystem is full.

Make sure that **xfs** is configured to run in the appropriate runlevels. Occasionally it may be necessary to delete stale lock, pid, or socket files. Once in a while, the font indexes in a font directory may be corrupt, and it will be necessary to run **mkfontdir** to recreate them. When this happens, **xfs** may seem to start correctly, but then dies. Try commenting out font paths in `/etc/X11/fs/config`, then run **xfs** from a terminal to determine which directory has problems.

Remember that X is a network service, even when you have not enabled access to your display. Unsuccessful hostname resolution can produce various behaviors, including the inability to launch applications from the panel. Changing hostnames can cause similar problems; if you need to change your computer's hostname, switch out of runlevel 5 and make sure X is not running (through **startx**), change the hostname, and only then restart X.

## Things to Check: Networking

- Hostname resolution
  - `dig www.redhat.com`
- IP configuration
  - `ifconfig`
- Default gateway
  - `route -n`
- Module specification
- Device activation

17-6

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

Hostname resolution problems can create problems for clients and servers alike. Aside from requiring successful forward lookups, reverse lookups are essential for many host-based security mechanisms. Tools like **host** and **dig** are invaluable for determining whether hostname resolution problems exist.

IP configuration may be checked using the **ifconfig** command, which will print information such as an interface's IP, the subnet mask, and other important settings. The **netstat -r** and **netstat -rn** commands will show if a system's routing table is correct. The absence of a default gateway or the existence of multiple default gateways can create problems. Inability to contact the default gateway (and thus, to reach the gateway to get outside the local network) can also cause networking problems.

It is possible that the kernel module for your particular network interface card has been mis-specified. For example, the Red Hat Enterprise Linux installer sometimes probes a de4x5-based card as a tulip-based card. Unfortunately, the tulip module will only work enough to enable the interface, but not enough to work.

Do not overlook the obvious: maybe the interface has not been activated, or was deactivated for some reason.

# Order of the Boot Process

- Bootloader configuration
- Kernel
- **/sbin/init**
  - Starting init
- **/etc/rc.d/rc.sysinit**
- **/etc/rc.d/rc**, **/etc/rc.d/rc?.d/**
  - Entering runlevel X
- **/etc/rc.d/rc.local**
- X

17-7

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In order to troubleshoot boot time problems, one must understand the boot process itself, remember how things look when they are working correctly, and narrow down how far into the process a failure is occurring.

*Issue:* No bootloader splash screen or prompt appears. *Possible Causes:* GRUB is misconfigured. Boot sector is corrupt. A BIOS setting, such as disk addressing scheme, has been modified since the boot sector was written.

*Issue:* Kernel does not load at all, or loads partially before a panic occurs. *Possible Causes:* Corrupt kernel image. Incorrect parameters passed to the kernel by the bootloader.

*Issue:* Kernel loads completely, but panics or fails when it tries to mount root filesystem and run **/sbin/init**. *Possible Causes:* Bootloader is misconfigured. **/sbin/init** is corrupted or **/etc/inittab** is misconfigured. Root filesystem is damaged and unmountable.

*Issue:* Kernel loads completely, and **/etc/rc.d/rc.sysinit** is started and interrupted. *Possible Causes:* **/bin/bash** is missing or corrupted. **/etc/fstab** may have an error, evident when filesystems are mounted or fsck'ed. Errors in software RAID or quota specifications. Corrupted non-root filesystems (due to a failed fsck).

*Issue:* Run level errors (typically services). *Possible Causes:* Another service required by a failing service was not configured for a given runlevel. Service-specific configuration errors. Misconfigured X or related services in runlevel 5

# Filesystem Problems During Boot

- **rc.sysinit** attempts to mount local filesystems
- Upon failure, user is dropped to a root shell
  - **fsck** may be used to fix corrupted filesystems
- Before running **fsck**:
  - Check **fstab** for mistakes
    - Run **mount -o remount,rw /** before editing
  - Manually test mounting filesystems

17-8

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

One of **rc.sysinit**'s jobs during the boot sequence is to mount most of the filesystems listed in **/etc/fstab**. If a filesystem appears not to have been unmounted correctly (for example, if the system suddenly loses power) **rc.sysinit** will run the **fsck** command on it.

**fsck** is a front end to the standard filesystem checking programs on the system. So for an ext2/ext3 filesystem, **fsck** actually executes the **e2fsck** utility. This tool repairs ext3 filesystems by using a special record on the filesystem called its *journal* and ext2 filesystems by exhaustively examining the filesystem's contents and metadata. The journal makes corruption of ext3 filesystems much less likely and speeds the process of checking a filesystems for errors. Journaling is one of the major advantages of ext3 over ext2.

If **rc.sysinit** is unable to mount a filesystem for any reason it will halt the boot process with the root partition mounted read-only and all other partitions unmounted, drop to a root password prompt and advise the user to run **fsck**. Be warned that **rc.sysinit** can be a little over-zealous in this regard. There are reasons for a filesystem to be un-mountable that have nothing to do with filesystem corruption, such as errors in **/etc/fstab** or a corrupted raid device. Since filesystem corruption is relatively rare with ext3 and since running **fsck** under the wrong circumstances can actually damage the filesystem, be sure to check **fstab** and attempt to manually mount each filesystem before running **fsck**.

If a mistake is discovered in **/etc/fstab**, you will need to remount the root filesystem with read-write permissions before you will be able to correct it. To do this, run the following command:

```
# mount -o remount,rw /
```

# Recovery Run-levels

- Pass run-level to init
  - on boot from GRUB splash screen
  - from shell prompt using: **init** or **telinit**
- Runlevel 1
  - Process `rc.sysinit` and `rc1.d` scripts
- Runlevel s, S, or single
  - Process only `rc.sysinit`
- emergency
  - Run **sulogin** only

17-9

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email [ctraining@redhat.com](mailto:ctraining@redhat.com) or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

In recovery situations, it is often helpful, (and depending on the problem possibly necessary) to boot to a run-level where services are inactive. For example, consider if you have a service that causes the machine to panic each time it tries to start. In this case, the road to recovery starts by preventing the service from starting, so you can successfully boot the machine to a stable state and determine the problem with the service. The below listed runlevels are of particular importance in system recovery situations.

## Runlevel 1

Booting to runlevel 1 will cause the system to process the `/etc/rc.sysinit` script followed by each of the `/etc/rc.d/init.d` scripts called in `/etc/rc1.d/*`. By default, Red Hat Enterprise Linux will only call the single script in this runlevel, which after some basic checks and cleanup will exec `init S`.

Switching to runlevel 1 from some other runlevel (3, 5, etc.) is a convenient way to kill all daemons as each of the `/etc/rc1.d/*` kill scripts will be processed.

## Runlevel s, S, single

Booting to runlevel single will cause the system to process the `/etc/rc.sysinit` script (if `/etc/inittab` is intact). If `/etc/inittab` is missing or corrupt, you can still boot to single mode, and in that case, you are given the root shell with no scripts processed.

Sometimes going to single user mode is overkill: interactive startup mode, invoked by typing "T" when "Welcome to Red Hat Enterprise Linux" appears at boot time, allows you to choose which services will run.

## Runlevel emergency

While technically not a runlevel, emergency mode shares many characteristics of the above listed runlevels. You can only access emergency mode during boot by passing emergency as a parameter from the grub prompt. No scripts will be processed, and you are given a root shell.

# Rescue Environment

- Required when root filesystem is unavailable
- Non-system specific
- Boot from CDROM (boot.iso or CD #1)
- Boot from diskboot.img on USB key

17-10

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

If the root filesystem is available and mountable, then you should be able to use it to fix problems that may occur. When it is not, then you must use a rescue environment. A rescue environment is a streamlined Red Hat Enterprise Linux system that does not require the installed OS to run. Rather than working on the broken system itself, you work outside of the system in an environment that, while more limited than single user mode (or even sulogin mode), should provide enough tools to recover root.

There are several ways to boot into the rescue environment:

Boot from CDROM, then type **linux rescue** at the isolinux prompt

Boot from a `diskboot.img` USB drive, then type **linux rescue** at the prompt

# Rescue Environment Utilities

- Disk Maintenance Utilities
- Networking Utilities
- Miscellaneous Utilities
- Logging: `/tmp/syslog` or `/tmp/anaconda.log`

17-11

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The rescue environment exists within a ramdisk image (referenced as `/dev/root`). Because of limitations on size and the number of inodes, many familiar utilities and device nodes are not available. However, tools related to disk maintenance (the probable reason for being in the rescue environment) and network connectivity are provided.

The following is a partial list of utilities provided by the rescue environment:

Disk Maintenance Utilities, including: a complete set of LVM utilities, for managing physical volumes, volume groups, and logical volumes; software RAID tools; swap commands; disk partition utilities; filesystem creators, checkers, debuggers, and labelers for `ext2`, `ext3`, `jfs`, `msdos`, `vfat`, and `reiser` filesystems.

Networking Utilities, including: network debuggers (**`ifconfig`**, **`route`**, **`traceroute`**, **`host`**); network connectivity tools (**`ftp`**, **`scp`**, **`ssh`**).

Miscellaneous Utilities including: shell commands (**`bash`**, **`chroot`**); process management tools (**`ps`**, **`kill`**, **`killall`**); editors (**`vi`**, **`nano`**); `mt` tools commands; kernel module management commands; archiving and compression tools (**`dd`**, **`tar`**, **`cpio`**, **`gzip`**); `rpm`; file manipulation commands (**`cd`**, **`ls`**, **`mkdir`**, **`cp`**, **`mv`**, **`rm`**)#.

Within the rescue environment, system logging information can be found in the file `/tmp/syslog`. Booting information is in `/tmp/anaconda.log`. Some configuration files (`modprobe.conf`, `netinfo`, and device files (`[sh]da`, `loop0`) are located in `/tmp` as well.



## Rescue Environment Details

- Filesystem reconstruction
  - Anaconda will ask if filesystems should be mounted
  - `/mnt/sysimage/*`
  - `/mnt/source`
  - `$PATH` includes hard drive's directories
- Filesystem nodes
  - System-specific device files provided
  - **mknod** knows major/minor #'s

17-12

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <training@redhat.com> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

The rescue environment will attempt to reconstruct the hard disk's filesystem under the mount point `/mnt/sysimage`. Since the rescue environment is often used on systems with damaged or misconfigured filesystems, however, this operation might or might not work. A corrupted partition table will appear to hang the rescue environment (a shell with `fdisk` is available under *Alt-F2*, however.) Using **linux rescue nomount** as the boot prompt directive disables automatic mounting of filesystems and circumvents the hanging caused by bad partition tables. Careful inspection of the output of the `mount` command should determine the state of the reconstructed filesystem.

Because the standard installation provides device node management through `udev`, administrators seldom need to create device nodes directly. In the rescue environment, device nodes are only provided for the most basic devices, including any fixed disks the kernel was able to auto-detect.

In order to access any other devices, such as a floppy drive, the relevant device node must be created with **mknod**. Fortunately, the rescue environment's version of **mknod** automatically associates the appropriate device driver major/minor numbers with well-known device names. For example, the device node for the master hard disk on the secondary IDE controller can be created with **mknod /dev/hdc**.

## End of Unit 17

- Questions and Answers
- Summary
  - X: Check **xfs** and full or over-quota filesystems
  - Networking: Check name resolution, routing, and device activation
  - Boot problems: Remember the sequence of events in the boot sequence
  - Repair options: Runlevels S, 1, and emergency. Rescue environment if those fail too.

17-13

For use only by a student enrolled in a Red Hat training course taught by Red Hat, Inc. or a Red Hat Certified Training Partner. No part of this publication may be photocopied, duplicated, stored in a retrieval system, or otherwise reproduced without prior written consent of Red Hat, Inc. If you believe Red Hat training materials are being improperly used, copied, or distributed please email <[training@redhat.com](mailto:training@redhat.com)> or phone toll-free (USA) +1 (866) 626 2994 or +1 (919) 754 3700.

# Lab 17

## System Rescue and Troubleshooting

---

Goal: To build skills in system rescue procedures.

Estimated Duration: 2 hours

## Sequence 1: Repairing the MBR in the rescue environment

**Scenario:** The rescue environment provides a last resort for repairing an unbootable machine, even when the bootloader or the root filesystem is damaged or misconfigured. In order to access the rescue environment, you will need either a `boot.iso` cdrom on a network that has the Red Hat installation tree (the `RedHat` directory) available via NFS, or a Red Hat Enterprise Linux CDROM.

### Instructions:

1. Use the following command to overwrite the first stage of GRUB in your Master Boot Record with zeros. Specify the block size carefully. If you write too many zeros, you will overwrite your partition table as well, and this will become a much more difficult exercise. (Note that the command below assumes you are using IDE drives. You might need to modify the destination device.)

After typing the following command, check it three times and hit *Enter* but once.

```
# dd if=/dev/zero of=/dev/hda bs=256 count=1 && reboot
```

You have just wiped out your boot sector, but your primary partition table will still be intact. Attempt a reboot to confirm that your system is unbootable. Use the Red Hat rescue environment to repair the system.

## Sequence 2: Installing software in rescue mode

### Instructions:

1. Use the following command to overwrite the **mount** command.

```
# cp /bin/date /bin/mount
```

You have just wiped out a key executable on your system. Upon attempting a reboot, you should find your system unbootable. Use the Red Hat rescue environment, along with its version of the rpm command and the library of RPMs provided by the installation tree, to repair the system. Hint: **/bin/mount** is part of the `util-linux` RPM.

## Sequence 3: Troubleshooting Practice

### System Setup:

1. Turn off iptables and mount the /var/ftp/pub directory from server1 if it is not currently mounted.

```
service iptables stop
chkconfig iptables off
mkdir /mnt/server1
mount server1:/var/ftp/pub /mnt/server1
```

2. Install the Troubleshooting Practice RPM:

```
rpm -ihv /mnt/server1/gls/RPMS/rhce-ts-*
```

3. Ensure that your computer is configured as closely as possible to the following specifications:
  - Authenticate users from your local /etc/passwd file. That is, do not run any network authentication scheme such as NIS or LDAP.
  - Use 192.168.0.254 (server1.example.com) as your name server.
  - Confirm that /usr/local/bin is part of your PATH environment variable.

The following items are required for some, but not all, troubleshooting problems. You may still do most problems if some of these items are missing.

- Change to runlevel 3, not runlevel 5. Confirm that the X server is not running (no startx). Only the local problems require this.
- Confirm that /home is a separate filesystem from the root filesystem and is local to the system (not an NFS mounted filesystem).

### Instructions:

1. The Troubleshooting Practice problems come in three parts, each invoked by a separate command. The sections, commands, and number of problems in each section vary; therefore, run **command count**, to determine the number of problems for each troubleshooting command:
  - For Local: **tslocal count**
  - For Services: **tsservices count**
  - For Networking: **tsnetwork count**
  - For Booting: **tsboot count**

2. Invoke the first local problem by running:

```
# tslocal 1
```

This command will set up the problem and will explain the goal. The goal will be stored in the file `/etc/ts` for later reference. Spend three to eight minutes trying to solve the problem.

3. If you have not yet solved the problem, you may need a hint. Hints can be displayed by running the **tshint** command:

```
# tshint local 1 1
```

This will display the first hint for the first tslocal problem. Continue to invoke hints until you get enough information to solve the problem or until you run out of hints:

```
tshint local 1 2
tshint local 1 3
[ and so on ...]
```

The tshint command will tell you when you have reached the end of the hints. Again, do not spend more than five to ten additional minutes on this problem.

4. Whether or not you have solved the problem, run the **tslesson** command:

```
# tslesson local 1
```

This command will tell the lessons intended to be taught by the problem. Some **tslesson** messages also give step-by-step instructions on how to approach a particular problem.

5. If, after reading the hints and the lesson, you are unable to solve the problem, call the instructor for assistance.

6. Proceed with the remaining problems in the same way. For example, **tsnetwork 1** sets up the first network problem and **tshint network 1 1** shows the first hint for the first network problem.

## Sequence 1 Solutions

1. Use the Red Hat rescue environment to repair the system.
  - a. Load the rescue environment by booting from a Red Hat installation media (either CDROM or PXE ), and typing `linux rescue` at the boot prompt. Proceed with the normal installation defaults. Choose NFS image for the media type and use the following NFS information:
    - NFS server : `server1.example.com`
    - NFS directory : `/var/ftp/pub`
  - b. The rescue environment will ask if you wish to mount the hard drive's filesystems. Select **Continue** to mount the filesystems in read-write mode. Examine the output of `mount` to confirm that the filesystem was correctly reconstructed. You might want to refresh your memory by examining your disk's partitions with `fdisk` .
  - c. Note that your hard drive has been reconstructed under the mount point `/mnt/sysimage`. Examine `grub.conf` (on your hard drive) to confirm that it is appropriately configured.

```
# cat /mnt/sysimage/boot/grub/grub.conf
```
  - d. To reinstall GRUB, you must shift contexts, so that `grub-install` believes that the root of your filesystem is the `/mnt/sysimage` directory. Spawn a chrooted shell, run `grub-install`, and then exit.

```
# chroot /mnt/sysimage
# grub-install /dev/hda
# exit
```

(Or, should the above fail to execute properly)

Exit the chroot environment and then type the command: **grub** at the bash prompt. This will place you into grub's command shell where you can enter the following commands:

```
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
```
  - e. Now exit your rescue shell. Note that the rescue environment will unmount any partitions that you mounted. Eject the CD.



## Sequence 2 Solutions

1. Use the Red Hat rescue environment, along with its version of the rpm command and the library of RPMs provided by the installation tree, to repair the system.
  - a. Load the rescue environment as in the previous exercise.
  - b. The rescue environment will attempt to automatically mount the hard drive's filesystems. Examine the output of mount to confirm that the filesystem was correctly reconstructed.
  - c. Verify the util-linux rpm on your hard drive, using a chrooted invocation of rpm. Do not forget to exit the chroot or the rpm installation will fail.

```
# chroot /mnt/sysimage
# rpm -V util-linux
# exit
```

- d. rpm should report that the /bin/mount executable has been modified. Reinstall the util-linux RPM from your installation tree (which has been NFS mounted under /mnt/source).

```
# cd /mnt/source/RedHat/RPMS
```

```
# rpm -ivh --force --root /mnt/sysimage util-linux*
```

Note that the util-linux package was installed (the hash marks indicate this), although you may see some errors at the end of the process. As it turns out, this is harmless error, although in a production environment, you would want to test this out fully.

- e. Now exit your rescue shell. Note that the rescue environment will unmount any partitions that you mounted.

