CramSession
The Original Study Guide

Over
4 Million
Downloaded

LPI

LPIC1 - Junior Level Administration Part 2

102 102 102 102 102 102 102 102 102 102 102

Matthew Rice - Author
Greg Rice - Technical Editor

Steven Johnson - Managing Editor

Your Trusted
Study Resource for
Technical Certification

The **Most Popular**
Study Guide on the web

LPIC1 – Junior Level Administration Part 2 (102) Study Guide

Copyright © 2006 by PrepLogic, Inc.
Product ID: 01447
Production Date: September 28, 2006

Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no warranty or fitness is implied. The publisher and authors assume no responsibility for errors or omissions. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

Volume, Corporate, and Educational Sales

PrepLogic offers favorable discounts on all products when ordered in quantity. For more information, please contact PrepLogic directly:

1-800-418-6789
solutions@preplogic.com

Abstract

This study guide will help you prepare for the Linux Professional Institute LPI 102 exam. This exam is one of two exams (the other is LPI 101) that candidates must pass in order to be awarded the LPIC-1 certification (level 1 Linux administration). This exam contains between 60 and 90 multiple-choice and fill-in-the-blank questions. Candidates have up to 2 hours to complete the exam.

What to Know

The objectives covered in the LPI 102 exam are:

- Topic 105: Kernel
- Topic 106: Boot, Initialization, Shutdown and Runlevels
- Topic 107: Printing
- Topic 108: Documentation
- Topic 109: Shells, Scripting, Programming and Compiling
- Topic 111: Administrative Tasks
- Topic 112: Networking Fundamentals
- Topic 113: Networking Services
- Topic 114: Security

Tips

- The LPI exams cover a broad range of Linux administration topics. It is useful to study all of the areas while keeping in mind that the weighting of each objective indicates how many of the exam questions will be devoted to the objective.

There is no substitute for hands-on experience.

Table of Contents

Topic 105: Kernel

1.105.1 Manage/Query kernel and kernel modules at runtime (Weight: 4)

Candidates should be able to manage and/or query a kernel and kernel loadable modules.

The Linux Kernel

The Linux Kernel is responsible for handling many tasks, including memory management, program loading and process scheduling, and providing access to hardware for applications.

Kernel versions are numbered major.minor.patch. If the minor version number is odd, the kernel is considered a development or testing version. If the minor version number is even, the kernel is considered a production version.

You can determine the version of kernel with the command:

```
[matt@pitt:~] uname -r
2.6.15-26-386
```

Also, you can find out more information about your system using the –a option to uname, for example:

```
[matt@pitt:~] uname -a
Linux otoole 2.6.15-26-386 #1 PREEMPT Mon Jul 17 19:52:53 UTC 2006 i686 GNU/Linux
```

Numbers after the '-' character indicate package information from your Linux distribution.

Kernel Modules

Modules are stored in the directory /lib/modules/`uname -r`, where 'uname -r' is the version of the kernel.

Linux uses a modular kernel; thus, unneeded functionality need not be included in the running kernel and using extra memory. Additionally, as you need more functionality, you can simply load a module to provide that functionality while you need it.

Loading and Unloading Kernel Modules

To install a simple module, use the insmod(8) command. To load support for fat file systems, use something such as:

```
[root@pitt:~] insmod /lib/modules/`uname -r`/kernel/fs/fat/fat.ko
```

On some distributions, you need not specify the full path to the module.

To unload a module, use the rmmod(8) command. You can now refer to the module by name instead of having to provide the full path.

```
[root@pitt:~] rmmod fat
```

Handling Module Dependencies

Due to the modular nature of the Linux Kernel, sometimes you cannot load a module because it depends on functionality from another module. For example, if you wanted to use a vfat or fat32 file system and try to simply load that module, your results would be:

```
[root@pitt:~] insmod /lib/modules/`uname -r`/kernel/fs/fat/vfat.ko
insmod: error inserting '/lib/modules/2.6.15-26-386/kernel/fs/vfat/vfat.ko': -1 Unknown symbol in module
```

To alleviate the problem presented here, the modprobe(8) command. modprobe will load module dependencies before the actual module that is required. With the vfat module example, you can perform the following:

```
[root@pitt:~] modprobe vfat
```

To confirm that the module was loaded, verify it with the lsmod(8) command. The following command shows that both the fat and vfat modules were loaded:

```
[root@pitt:~] lsmod |grep fat
vfat         13440 0
fat         53020 1 vfat
```

1.105.2 Reconfigure, build and install a custom kernel and kernel modules (Weight: 3)

Candidates should be able to customize, build and install a kernel and kernel-loadable modules from source.

Customize the current kernel configuration. Build a new kernel and appropriate kernel modules. Install a new kernel and any modules. Ensure that the boot manager can locate the new kernel and associated files.

Configuring the Kernel

The typical location for unpacking your kernel source code is in /usr/src. A new directory will be created as /usr/src/linux-<version>. The /usr/src/linux directory should be a symbolic link to the kernel source (also in /usr/src) against which you are trying to compile additional software. Under normal circumstances, you will not need a /usr/src/linux directory.

Once you have unpacked your source code, change directory to the top directory of the source code. You will have a number of configuration commands available through the use of the make(1) command and the Makefile file at the top of the kernel source directory tree. Current configuration options include:

Configuration Target	**Description**
make config	This is the most foolproof (but tedious) option. A long series of Yes/No questions will be asked; if there are any mistakes, you have to start over.
make menuconfig	A text-based configuration tool with some simple and useful navigation for choosing your options.
make xconfig	A QT-based configuration front end.
make gconfig	A GTK-based configuration front end.
make oldconfig	Update current config using a provided .config as a base.

Once you have successfully created a configuration, the options are saved in the file .config. Protect this file. There is no backup and the file can easily be deleted.

Topic 106: Boot, Initialization, Shutdown and Runlevels

1.106.1 Boot the system (Weight: 3)

Candidates should be able to guide the system through the booting process.

The boot process includes the following steps:

1. System power comes on and the computer goes through the POST step (Power-On Self Test).

2. The system loads a BIOS (Basic Input/Output System). The BIOS initializes hardware. Its its final responsibility is to load something from a boot sector on another device and pass on execution to the code that is loaded from the boot sector.

3. The boot sector contains a Boot Loader (such as GRUB or LILO). This boot sector can be the first sector on a floppy disk, USB memory stick, CD-ROM or a hard drive. On a hard drive, the first sector is called the Master Boot Record (MBR); it also includes partition information.

4. The Boot Loader's responsibility is to find either another Boot Loader or, more commonly, a Linux Kernel to load. At this point, the user may be able to provide information that will be passed to the Linux Kernel to change hardware settings, the runlevel and many other settings.

5. Once a Linux Kernel is chosen (or the default is selected or the time period for user choices has expired), the Boot Loader will load the Linux Kernel into memory and pass on execution to the Linux Kernel.

6. The Linux Kernel will initialize devices, load an initial RAM disk (initrd) and any required modules. Its final steps are to mount the root file system (which may be overridden with the root=/dev/hdXXX parameter) and load and execute the /sbin/init program (which may also be overridden with the init=/some/program parameter).

7. The /sbin/init program becomes the first process on the running Linux operating system and has a PID (Process ID) of 1.

8. /sbin/init will read its configuration file /etc/inittab to determine the default runlevel unless a runlevel was passed to the Linux Kernel during the Boot Loader stage. The line in /etc/inittab that determines the default run level looks like the following:

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

The numeric value is the default run level.

9. The init(8) process will then run all programs that are configured for the chosen run level. These are also defined in the /etc/inittab file. Some of these programs are getty programs (such as mingetty) which provide console logins and graphical login programs such as xdm, gdm and kdm.

Runlevel Directories

Most of the services made available by a Linux system are through “init” scripts that are run due to an entry in the /etc/inittab file. The exceptional entries in which we are interested look like the following:

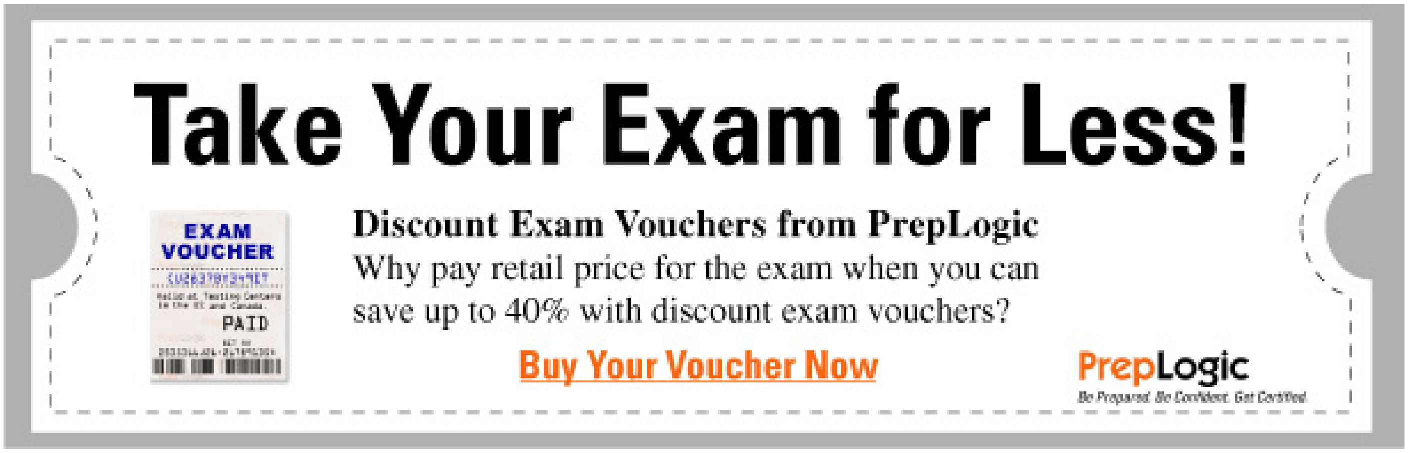
```
[matt@pitt:~] grep "rc [0-6]" /etc/inittab
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

The entry that specifies a 2 in the runlevel field means that, when init(8) enters runlevel 2, the script /etc/init.d/rc is executed with a 2 passed as the only argument.

You can also start these services manually by running them with the start option, as follows:

```
[root@pitt:~] /etc/init.d/exim start
```

Other useful options are “stop,” “restart” and “status.” Most scripts will support all these options.



The numbers after the “S” tell the /etc/init.d/rc script the order in which to execute these startup scripts. The lower-numbered scripts are executed first. These scripts can perform numerous tasks, such as start a web or e-mail server, run a graphical login (some distributions run the graphical login right in /etc/inittab, though) or configure hardware.

RedHat places all of the /etc/rcN.d directories in /etc/rc.d. If you inspect /etc/rc2.d with ls, you will see that it is a link into /etc/rc.d:

```
[matt@pitt:~] ls -l /etc/rc2.d
lrwxrwxrwx 1 root root  10 Sep 18 10:20 /etc/rc2.d -> rc.d/rc2.d
```

Although the placement is confusing, the rc script is also in the location /etc/rc.d/rc. Inspect /etc/inittab on a RedHat system to see the differences. You must know both the RedHat and Debian layouts for the exam.

Managing Runlevel Services

The scripts in /etc/rcN.d are symbolic links back to an original script in the /etc/init.d directory. This arrangement ensures that users do not copy the same script into each runlevel. Numerous utility programs are available to help you manage these links; the programs include chkconfig, update-rc.d and ntsysv.

The startup scripts sitting in the /etc/init.d directory are referred to as init scripts.

Managing Runlevel Services with ntsysv

The program ntsysv is a graphical tool. If you run it without any arguments, you can configure your current runlevel. If you want to specify alternative runlevels to configure, run ntsysv with the –level option such as:

```
[root@pitt:~] ntsysv --level 1
```

to configure runlevel 1. Or:

```
[root@pitt:~] ntsysv --level 23
```

to configure multiple runlevels at the same time.

When ntsysv is run, you will be presented with a list of available services. Press the space bar to toggle the service on or off. An asterisk in the square brackets ([*]) indicates that the service is on. No asterisk means that the service will not execute for this runlevel.

Managing Runlevel Services with chkconfig

To list the services and their applicable runlevels with chkconfig, use the --list option. The following listing shows a sample of the output:

```
[root@pitt:~] chkconfig --list
pcmcia    0:off  1:off  2:on  3:on  4:on  5:on  6:off
nfs-common   0:off  1:off  2:off  3:on  4:on  5:on  6:off
xprint    0:off  1:off  2:off  3:on  4:on  5:on  6:off
setserial   0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

or if you are interested in a specific service:

```
[root@pitt:~] chkconfig --list nfs-common
nfs-common   0:off  1:off  2:off  3:on  4:on  5:on  6:off
```

The output of this command shows the various services that are installed and lists the runlevels. Any runlevel with the word on after it indicates that there is a startup script in the appropriate /etc/rcN.d directory. The letter N represents a runlevel number.

To modify the runlevels that the /etc/init.d/rc script starts, use a command such as:

```
[root@pitt:~] chkconfig --level 345 nfs-common on
```

The options “on,” “off” and “reset” are available.

The chkconfig program will also inspect startup scripts in /etc/init.d (or /etc/rc.d/init.d for older RedHat-based systems) for special comments to indicate default runlevels. If these comments are in the file and the suggested levels meet your requirements, you can add it to these runlevels with:

```
[root@pitt:~] chkconfig --add nfs-common
```

LILO

The LILO (Linux Loader) boot loader is configured through the use of the /etc/lilo.conf file. This file is generally broken into two main sections; global and per-image options. Some per-image options are further separated depending on whether they are for a Linux kernel or an arbitrary system. Additionally, many of the per-image options may be used at the global level to indicate a default value.

Essential Global Options for /etc/lilo.conf

Many options may be placed in /etc/lilo.conf. Only the most common ones are listed here.

Option	Description
boot=<device>	This setting specifies the name of the device that contains the boot sector. In this example, the first sector (or MBR) of the first IDE hard drive is used as the boot sector. If you wanted the LILO boot loader placed into a partition on /dev/hda, you would specify a specific partition such as /dev/hda3.
default=<label>	This setting specifies the default kernel image that will boot. If this option is omitted, the first image listed in /etc/lilo.conf will be used as the default.
timeout=<tsecs>	This setting specifies the amount of time (in tenths of a second) that LILO will wait for keyboard input before booting the default kernel image. You must use the prompt option to enable the timeout. A setting of 150 indicates a 15 second timeout.
prompt	This option instructs LILO to issue the boot: prompt and wait for user input.
lba32	This option enables LILO to boot from disks where the kernel image resides on a partition that is past the 1,024th cylinder.
vga=<value>	This selects the VGA text mode that is used when booting. Values for this setting are "normal," "extended," "ask" or a number.
read-only	This option indicates that the root file system should be mounted read-only. Usually, the operating system will remount the file system to read-write.

GRUB

The GRUB (Grand Unified Boot Loader) boot loader was the first loader to boot Linux from above the 1024th cylinder of a hard drive. GRUB has taken over as the default boot loader for many Linux distributions because it offers many features that LILO lacks. For example, you need not reinstall the GRUB boot loader after editing its configuration, and you have many more interactive options while booting.

Configuring the GRUB Boot Loader

The usual location for the GRUB boot loader's configuration file is /boot/grub/menu.lst. On some systems the configuration file is located at /boot/grub/grub.conf. GRUB can read its configuration file at boot time because it supports many different file systems.

Essential Global Options for /boot/grub/grub.conf

The following table lists common global options used in a GRUB configuration file.

Option	Description
default=<value>	This option tells GRUB which operating system in the configuration file to boot as the default (GRUB indexes from 0). To boot the second listed operating system, use default=1.
timeout=<secs>	This option defines how long, in seconds, to wait for user input before booting the default operating system. A setting of 15 indicates a 15-second timeout.

The GRUB timeout option is in seconds while the LILO timeout option is in tenths (1/10) of a second.

1.106.2 Change runlevels and shutdown or reboot system (Weight: 3)

Candidates should be able to manage the system's runlevel. This objective includes changing to single-user mode, shutting down or rebooting the system. Candidates should be able to alert users before switching runlevel and properly terminate processes. This objective also includes setting the default runlevel.

Set the default runlevel

To set the default run level, change the numeric value in /etc/inittab on the line that looks like the following:

```
id:2:initdefault:
```

Values between 1 and 5 are the most useful. A standard exists for the meaning of each runlevel defined by the LSB (Linux Standard Base) but not every distribution follows the standard.

Change between run levels

Use the runlevel command to determine your current run level. The output will look something like the following:

```
[matt@pitt:~] runlevel
N 5
```

The first character indicates the previous runlevel. An "N" indicates that the system booted directly into a runlevel and has not been changed since boot time.

The second character is the current runlevel (in this example, that is runlevel 5).

To change to another runlevel, use the init(8) or telinit(8) programs. For example, to switch to runlevel 3, use the command:

```
[root@pitt:~] init 3
```

This command will send a message to the currently running init process (PID 1), which will subsequently reread the /etc/inittab file, remove any processes that are no longer required for the runlevel and call the /etc/init.d/rc script with the new runlevel to add or remove even more services.

Shutdown and reboot from the command line

The shutdown(8) command is the most versatile command for shutting down the system. A sample use of the command would look like the following:

```
[root@pitt:~] shutdown -h 19:00 "system coming down for a new disk"
```

This example would switch the system to runlevel 0 (halt) at 7 p.m. that night (or the next night if the command is executed later than 7 p.m.) and send the message “system coming down for a new disk” to all logged-in users. Further login attempts will also be blocked.

Shutdown performs its functions by calling init(8) for the runlevel change.

The time may be given in the hh:mm format (hh is hours and mm is minutes on a 24-hour clock), in the format +m (m is the number of minutes to wait) or the word “man” (an alias for +0).

Topic 107: Printing

1.107.2 Manage printers and print queues (Weight: 1)

Candidates should be able to manage print queues and user print jobs.

A print queue is a spool area for files being printed.

A print job is an instance of a file in the print queue.

Two printing systems are in common use, LPD and CUPS. CUPS is the newer, better technology but there is a compatibility layer where the LPD commands are supported.

The LPD Commands

LPD commands come in two flavors, BSD and SysV. In the table below, the SysV versions are in parentheses. You should completely understand the BSD commands.

Command	Purpose
lpr (lp)	Used to submit files to a print queue
lprm (cancel)	Used to cancel print jobs
lpq (lpstat)	Displays the print jobs in a queue

The most commonly used option to these commands is “-P destination,” which indicates the named print queue that will receive the print job (lpr), cancel request (lprm) or queue query (lprm).

The default print queue is “lp” but this default can be changed with the PRINTER environment variable.

1.107.3 Print files (Weight: 1)

Candidates should be able to manage print queues and manipulate print jobs.

1.107.4 Install and configure local and remote printers (Weight: 1)

Candidates should be able to install and configure local and remote printers.

For print queues to work, the lpd(8) daemon must be running.

Refer to the printcap(5) man page for more details. For details on the formatting for /etc/printcap, please refer to the termcap(5) man page.

Entries in /etc/printcap are a single logical line with a '\' used at the end of each line to indicate a continuation to the next real line.

Configuring CUPS printers

CUPS is much more complex to configure. Only the basics are covered here. Normally all configuration of the CUPS printing system is done via the URL and web application at:

http://localhost:631/

This service is provided by the main CUPS server process, cupsd(8).

Important files for CUPS are:

/etc/cups/cupsd.conf	The configuration file for the CUPS scheduler, cupsd(8).
/etc/cups/classes.conf	Defines local printer classes that are available and is automatically generated by the cupsd(8) daemon when printer classes are added and removed.
/etc/cups/client.conf	Configuration for cups client programs. May also be personalized by placing in a ~/.cups directory for each user.
/var/spool/cups/	The default spool directory for print jobs.

Topic 108: Documentation

1.108.1 Use and manage local system documentation (Weight: 4)

Candidates should be able to use and administer the man facility and the material in /usr/share/doc/.

The info Command

An alternative documentation system to man pages is called info pages. Info pages use a custom reader for the documentation (as opposed to man using only the default pager). To learn how to use the info command, type:

```
[matt@pitt:~] info info
```

Other System Documentation

Package documentation that is not in the man or info formats are placed in the directory /use/share/doc. Package documentation is in the subdirectory <package>-<version> or <package>. Some documentation is also placed in /usr/doc.

1.108.2 Find Linux documentation on the Internet (Weight: 3)

Candidates should be able to find and use Linux documentation on the Internet.

This objective includes using Linux documentation at sources such as the Linux Documentation Project (LDP), vendor and third-party Web sites, newsgroups, newsgroup archives and mailing lists.

The home of the LDP is http://www.tldp.org. It is the official home of guides or tutorials on how to set up certain technologies called HOWTOs. Many other guides, FAQs (Frequently Asked Questions) and mini-HOWTOs are also available.

1.108.5 Notify users on system-related issues (Weight: 1)

Candidates should be able to notify the users about current issues related to the system.

This requirement means being familiar with the various files that contain messages as users log in.

File	Description
/etc/issue	Contains a message that is displayed above the login: prompt for local logins.
/etc/issue.net	Contains a message that is displayed above the login: prompt for remote logins (such as telnet). SSH doesn't use this file, though.
/etc/motd	The "message of the day" which is displayed after a successful login but before your shell is executed.

Topic 109: Shells, Scripting, Programming and Compiling

1.109.1 Customize and use the shell environment (Weight: 5)

Candidates should be able to customize shell environments to meet users' needs.

More topics relating to shells and their use are found in the LPI 101 exam.

Variable Substitution

You will frequently need to do more than simple variable assignment, especially when you want to use spaces in your variable values or the output from some command.

This task can be accomplished with one of three special quoting characters:

double quotation marks ("), the single quotation mark (') and the back tick (`).

Use double quotation marks (") around the value if you want variable substitution to occur in the value.

If you want to prevent the shell from treating a character as special, escape it with a '\'.

Exporting Variables

Normal variables are accessible only to the shell instance that created them. However, if you need a variable to be available to the scripts and programs that your shell executes, you must use the export command to indicate that the variable should be passed to the child process that you execute.

To export an existing variable, use a command such as:

```
[matt@pitt:~] export MYVAR
```

Also, you can do the variable assignment and export in one command, if that is more convenient. For example:

```
[matt@pitt:~] export MYVAR=SomeValue
```

You may also export (and assign) more than one value at a time:

```
[matt@pitt:~] export MYVAR=SomeValue
YOURVAR=SomeOtherValue
```

Inspecting Variables

To see a list of all your shell variables, use the set command:

```
[matt@pitt:~] set
<snip>
OPTIND=1
OSTYPE=linux-gnu
P4CONFIG=.p4settings.
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
<snip>
```

The set command will also list functions that you may have defined. Also, the set command has many other uses. Refer to the bash(1) man page for more details.

Removing Variables

There are two approaches to cleaning up variables that you no longer want.

One method is to assign an empty or “NULL” value to the existing variable, as follows:

```
[matt@pitt:~] MYVAR=
```

The variable will still exist but the value is an empty string.

To completely remove a variable, including its name, from your shell, use the unset command:

```
[matt@pitt:~] unset MYVAR
[matt@pitt:~] set | grep MYVAR
<no mention of MYVAR, if really gone>
```

Using Shell Functions

Shell functions are just like functions in other programming languages. Parameters can be passed, multiple statements may be executed, other functions may be called and values can be returned.

To define a function, enter something comparable to the following:

```
[matt@pitt:~] function my_top() {
        uptime
        free
        ps axuw | head -10
}
```

Precedence of Execution

The order of execution precedence for the bash shell is:

- Aliases - The shell will replace the alias with the alias value.
- Keywords - Such as if, for, while, while, until and function.
- Functions
- Built-in Commands - To save execution time, many useful commands are compiled into the shell interpreter such as alias, cd, echo. Confirm this compilation with the type command.
- Executables - Finally the PATH environment variable is checked to locate the command that is needed.

1.109.2 Customize or write simple scripts (Weight: 3)

Candidates should be able to customize existing scripts or write simple new BASH scripts.

A Simple Shell Script

A simple script looks like:

```
[matt@pitt:~] cat simple.sh
#!/bin/sh
# My 'hello world' shell script.
echo "hello $1"
```

The first line indicates to the kernel which shell interpreter to execute. That shell will read in this script and execute the commands.

The second line is a comment. Comments start with a “#” anywhere on the line, but not in quoted strings, and comment out the rest of the line.

The third line uses the echo command to print the string 'hello' followed by whatever (if anything) is the first argument to the script.

Executing the Script

To execute or run the simple.sh script, you can run a bash shell and give it the name of the script as an argument. Any further parameters will be passed to the shell script.

```
[matt@pitt:~] sh simple.sh world
hello world
```

Alternatively, you can turn on the read and execute permissions for the script and run it as you would any other script or program, as follows:

```
[matt@pitt:~] chmod +rx simple.sh
[matt@pitt:~] ./simple.sh world
hello world
```

We needed the ./ path information to find the simple.sh script because the home directory is not in the PATH environment variable.

Using Parameters in your Script

All parameters are numbered. To use a parameter higher than $9, you must use curly braces ({ and }) around the number.

The read Command

Users can also provide input with the read command. For example:

```
[matt@pitt:~] cat simple4.sh
#!/bin/sh
echo "provide two numbers and a sentence"
read var1 var2 theRest
echo $theRest: $(($var1 + $var2))
[matt@pitt:~] ./simple4
provide two numbers and a sentence
34 56 my age plus my IQ
my age plus my IQ: 90
```

In this script, the line “provide two numbers and a sentence” is printed to the screen. The user provides input and the two numbers are assigned to the first two variables (var1 and var2). Any remaining input is assigned to the last variable. If there are more variables than values, the remaining variables will be left unassigned.

The $((...)) construct is provided by bash for arithmetic calculations. More detail is provided in the bash(1) man page.

The test Command

The test command is used to compare file information (such as size, timestamps, type and permission), string and numeric comparison.

Scripting Control Constructs The if/elif/else/fi Statements

If statements allow for conditional execution based on the truthfulness of the value being tested. The arguments to the if family of statements is a command (including test or [). If the exit value is zero (0), the condition is considered true. Otherwise, the condition is false.

The for Loop

The for loop is used to iterate over a list of values. A simple usage is:

```
[matt@pitt:~] for i in 1 2 3 4 5; do
        echo "i is $i"
done
```

which produces the output:

```
i is 1
i is 2
i is 3
i is 4
i is 5
```

As with the if statement, you can put the do keyword on the next line, such as:

```
[matt@pitt:~] for i in 1 2 3 4 5
do
        echo "i is $i"
done
```

The while/until Loop

The while/until loops are provided to continue looping around a set of commands until the tested condition becomes true (an exit value of 0) in the case of the while loop, or false (an exit value of non-zero) in the case of the until loop.

Topic 111: Administrative Tasks

1.111.1 Manage users and group accounts and related system files (Weight: 4)

Candidates should be able to add, remove, suspend and change user accounts.

User accounts are assigned a User ID (UID) and one primary Group ID (GID) and any number of secondary Group IDs. UIDs and GIDs are numeric values from 0 to 65,536.

Creating New User and Group Accounts

User accounts can be created by manually editing the /etc/passwd and related files; however, that method is error prone. Instead, user accounts should be created with the useradd(8) command, such as:

```
[root@pitt:~] useradd newusername
```

On some distributions, this command will create a user entry in /etc/passwd and /etc/shadow and assign the user to a common group (such as the “users” or “staff” group). Other distributions will also create a group with the name of “newusername” and make that the primary group for the user.

Assigning Passwords

Once a user or group is created, you must set any required passwords. To change a user password, user the passwd(1) command, as follows:

```
[root@pitt:~] passwd newusername
```

You will be prompted for the new password information. Non-root users can only change their own password. In that case, the passwd(1) command requires no arguments.

Account status information can also be obtained with the '-S” option”

To administer the /etc/group file, use the gpasswd(1) command.

Modifying User and Group Accounts

To modify the user information such as the GECOS field, home directory, password and account expiry, group membership and more, use the usermod(8) command.

The chage(1) command can also be used to modify the user's password expiry information.

To modify a group's GID value or the group name, use the groupmod(8) command.

These commands can only be used by the root user.

Deleting User and Group Accounts

Use the userdel(8) command to delete a user. By default, the home directory, mail spool, and crontabs will be left intact. To delete the home directory, use the “-r” option. The administrator must still clean up mail and cron files as well as anything left behind in /tmp or elsewhere.

Use the groupdel(8) command to delete a group.

1.111.2 Tune the user environment and system environment variables (Weight: 3)

Candidates should be able to modify global and user profiles.

Using the /etc/skel Directory

Any files in /etc/skel will copied into a new user account's home directory.

Setting Environment Variables

Add environment variables for global users in the /etc/profile file because they are needed from the first login shell.

The ~/.bash_profile or ~/.profile files can be used for per-user settings.

Updating the PATH Environment Variable

The PATH environment variable is useful for finding programs on the system. For example, to add a bin/ directory in your home to the PATH environment variable, enter the following:

```
[matt@pitt:~] PATH="~/bin:$PATH"
[matt@pitt:~] export PATH
```

You need not export the PATH variable if it has already been exported.

1.111.3 Configure and use system log files to meet administrative and security needs (Weight: 3)

Candidates should be able to configure and manage system logs.

System logging with syslogd

Most system services running on a Linux Operating System will report logging information to the syslogd daemon. This daemon will then write the logs to files (typically in the /var/log directory), directly to the screen or even to another server on the network.

The default system log file is /var/log/messages. You can look through these log files for various events such as login attempts. For example, on an Ubuntu system, login attempts go to /var/log/auth.log:

```
[root@pitt:~] grep LOGIN /var/log/auth.log
Jul 17 20:41:50 otoole login[4900]: ROOT LOGIN on `tty1'
```

You can also use tail with the “-f” option to watch as new messages are logged to a syslog file.

Syslog Features

The syslogd daemon reads its configuration from the /etc/syslog.conf file. This file lists the logging functionality, selectors and actions. A selector is a combination of facilities and priorities. A facility is a category or class of message and the priority level varies from informational to emergency or critical.

The /etc/syslog.conf File

The following are some samples that may be found in the /etc/syslog.conf file.

Log all facility messages, except uucp, with a priority of warning to the /var/log/warning.log file. The '-' indicates that this file will not be synced after every write.

```
*.=warning;uucp.none      -/var/log/warning.log
```

All critical condition messages will be sent to the host 'loghost'.

```
*.crit                    @loghost
```

Log all but debug priority messages for mail, news and uucp facilities to any terminals that the root or matt users are logged on to.

```
mail,news,uucp.*          root,matt
```

Log all mail facility messages except emerg to the console.

```
mail.!=emerg              /dev/console
```

Security Log Files

Some other log files are kept in /var/log (and other directories) which are not managed by the syslogd daemon.

The who, w, last and lastlog commands query them for information.

The logrotate Program

The syslogd daemon has no facility for truncating or removing large log files. They will continue to grow.

To deal with this issue, the logrotate(8) command was created. It reads configuration information from /etc/logrotate.conf and the files in /etc/logrotate.d.

1.111.4 Automate system administration tasks by scheduling jobs to run in the future (Weight: 4)

Candidates should be able to use cron or anacron to run jobs at regular intervals and to use at to run jobs at a specific time.

The at Command Family

The at command is used to schedule a task to run once at some point in the future.

The cron Service

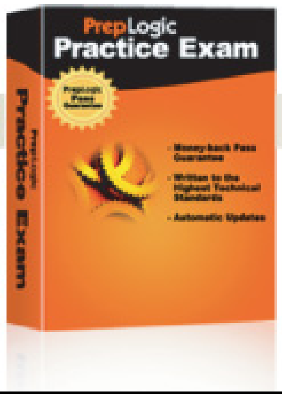
The cron service allows scheduling of tasks, as do at and batch. The chief difference with cron is that these are tasks that are expected to run periodically.

Any time that a cron job is run, any output created by the job will be mailed to the user. Also, if the job exits with a non-zero value, the user will receive an e-mail notification.

The list of cron jobs (or crontabs) are kept in the /var/spool/cron/ or /var/spool/cron/crontabs/ directory.

Use /etc/cron.allow to list users allowed access to the cron service. Use /etc/cron.deny to list users restricted from access to the cron service. If neither file exists, only root has access. If only one file exists, all missing users are assumed to have the opposite access to what the file provides.

Systemwide jobs can be placed in the /etc/crontab file. Also, hourly, daily, weekly and monthly jobs can be scheduled by placing scripts or programs in /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly and /etc/cron.monthly, respectively. These jobs will run as root.



The crontab Command

The list and scheduling of the cron jobs are managed with the crontab(1) command. The important options to crontab are:

Option	Description
-e	Edit the crontab.
-r	Remove the entire crontab.
-l	List the crontab.
-u	Specify a user other than the one running the command (only for root).

The anacron Command

An alternative system to cron is anacron. Anacron will schedule commands periodically but takes computer downtime into account. So, if a job would have run but the computer was shut off, the job will run when the system restarts.

The default configuration file is /etc/anacrontab and the spool directory is /var/spool/anacron.

1.111.5 Maintain an effective data backup strategy (Weight: 3)

Candidates should be able to plan a backup strategy and back up filesystems automatically to various media. The gzip, gunzip, bzip2, bunzip2, compress and uncompress commands provide compression and uncompression functionality.

1.111.6 Maintain system time (Weight: 4)

Candidates should be able to properly maintain the system time and synchronize the clock via NTP.

Setting System Time

The system time can be set with the date command such as:

```
[root@pitt:~] date 123115302006.05
```

This command will set the system clock to 3:30:05pm Dec 31, 2006. The format is MMDDhhmm[[CC]YY][.ss].

Use hwclock to read or assign the CMOS clock on the computer. To read the CMOS clock and assign it to the system time, use:

```
[root@pitt:~] hwclock --hctosys
```

To assign the system time to the CMOS (or hardware clock), use:

```
[root@pitt:~] hwclock --systohc
```

All time zone information is stored in files in the /usr/share/zoneinfo directory. To select a default time zone, link the appropriate file to /etc/localtime. For example, the following system is using Eastern Time (Eastern United States/Toronto, Canada) time zone.

```
[matt@pitt:~] ls -l /etc/localtime
lrwxrwxrwx 1 root root 35 12:43 /etc/localtime ->
/usr/share/zoneinfo/America/Toronto
```

Some systems also put the time zone string in /etc/timezone.

Using the NTP Service

NTP, the Network Time Protocol, can be used to set the system clock based on information provided by other time sources.

Topic 112: Networking Fundamentals

1.112.1 Fundamentals of TCP/IP (Weight: 4)

Candidates should demonstrate a proper understanding of network fundamentals.

Important Network Concepts

Term	Description
IP Address	Assigned statically or through a dynamic server such as DHCP.
Network Mask	The portion of the IP address used to determine the network address (as opposed to the client address).
Broadcast Address	The network address plus all client address bits turned on. Used to send a message to all hosts on a network at once.
Gateway Address	The address of the host, router or other device that can forward network packets from your local network to the next

An IP address is represented by 32 bits (4 8-bit bytes) for IPv4. The first (or higher order bits) indicate the network address while the lower order bits indicate the client address on that network.

An IP address of 192.168.2.12 and a network mask (or netmask) of 255.255.255.0 mean that the first 24 bits (3 8-bit bytes) are for the network address and the last 8 bits (1 8-bit byte) are for the client address.

Bits in an 8-bit byte have the values 128, 64, 32, 16, 8, 4, 2, 1 in left-to-right order. Add the values for each bit turned on to get the mask value (all bits on totals 255).

The address and network mask together can be specified as 192.168.2.12/255.255.255.0 or the shorter 192.168.2.12/24 (for a 24 bit network).

To compute the IP address, netmask, etc from the basic bits:

Network Addr: 11000000.10101000.00000010.10000000 (netmask==25 bits)

Client Addr: 00000000.00000000.00000000.01000000 (lower 7 bits)

Add the bits together in different combinations to get:

Network Addr:	11000000.10101000.00000010.10000000 (192.168.2.128)
Network Mask:	11111111.11111111.11111111.10000000 (255.255.255.128)
IP Address:	11000000.10101000.00000010.11000000 (192.168.2.192)
Broadcast:	11000000.10101000.00000010.11111111 (192.158.2.255)

In the above example, the last seven bits are used to provide client addresses, thereby producing 2^7 (or 128) addresses. However, all bits off means the network address and all bits on mean the broadcast. So, in fact, there are only 126 usable IP addresses on this subnet. They range from:

Network Addr:	11000000.10101000.00000010.10000000 (aka, /25)
Lowest Addr:	00000000.00000000.00000000.00000001
Highest Addr:	00000000.00000000.00000000.01111110

When the lowest and highest addresses are combined with the network bits, the addresses 192.168.2.129 and 192.168.2.254 are determined.

Networks are sometimes referred to by “class:”

Class	Description
Class A	A /8 or 255.0.0.0 netmask. Also, formally used by the addresses starting with the number 1 to 126. There are 126 Class A networks; each can have 16,777,216 (2^24) host addresses.
Class B	A /16 or 255.255.0.0 netmask. Also, formally used by the addresses starting with the number 128 to 191. There are 16,382 Class B networks; each can have 65,536 host (2^16) addresses.
Class C	A /24 or 255.255.255.0 netmask. Also, formally used by the addresses starting with 192 to 233. There are 2,097,150 Class C networks; each can have 254 (2^8 -2) host addresses.
Class D	From 224 to 239. This range is reserved for activities such as multicast and is not usually available for host addresses.
Class E	From 240 to 254. Reserved for future use.

Port Numbers

Sending a packet to a remote system is insufficient. To determine which application gets the data, that application must be listening on a “port” for connections and data. Ports are numbered and a list of assigned port numbers is in the /etc/services file.

Commonly used port numbers from /etc/services are:

Port	Assigned Use	Port	Assigned Use
20	FTP-DATA (for data transfer)	21	FTP (command channel)
22	SSH	23	TELNET
25	SMTP	53	DNS
80	HTTP	110	POP3
119	NNTP (USENET)	143	IMAP2
161	SNMP	443	HTTPS

Data Transport Protocols

Packets come in many formats depending on their purpose:

Type	Purpose
IP	Internet Protocol - The lower-level protocol upon which more complex protocols are based.
ICMP	Internet Control Message Protocol - This protocol defines a small number of messages used for diagnostic and management purposes.
UDP	User Datagram Protocol - An unreliable, connectionless protocol. Unreliable really means that there is no guarantee that a sent packet will arrive in order or even arrive.
TCP	Transmission Control Protocol - A reliable service that ensures that the receiving application is given the data in the order that it is sent. It is connection based.

To send an ICMP packet to another host (and receive an ICMP packet in response), use the ping(8) command:

```
[matt@pitt:~] ping -c 2 -n 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.048 ms
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.048/0.050/0.052/0.002 ms
```

No response means that the round trip was not successful. Something could be blocking the packet (although blocking is discouraged) or the remote machine may not be turned on.

The traceroute command is used to determine where along the network packets are being dropped as well as some performance indicators:

```
[matt@pitt:~] traceroute -n 10.10.10.2
traceroute to 10.10.10.2 (10.10.10.2), 30 hops max, 40 byte packets
 1 172.17.0.1 18.884 ms 20.327 ms 20.096 ms
 2 10.10.10.2 19.769 ms 22.522 ms 19.831 ms
```

To watch IP traffic passing on your network interfaces, use the tcpdump command:

```
[root@pitt:~] tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ath0, link-type EN10MB (Ethernet), capture size 96 bytes
16:34:10.390247 IP fibs.com.4321 > 192.168.2.192.38840: P 1420461721:1420461855(134) ack
3245961853 win 5792 <nop,nop,timestamp 1170903922 67293304>
16:34:10.573788 IP 192.168.2.192.38840 > fibs.com.4321: . ack 134 win 15948
<nop,nop,timestamp 67295730 1170903922>
...
```

1.112.3 TCP/IP configuration and troubleshooting (Weight: 7)

Candidates should be able to view, change and verify configuration settings and operational status for various network interfaces.

Viewing and Setting Network Interface Information

Use the ifconfig command to view all active interfaces:

```
[matt@pitt:~] ifconfig
eth0 Link encap:Ethernet HWaddr 00:05:4E:51:3C:44
   inet addr:192.168.2.192 Bcast:192.168.2.255 Mask:255.255.255.0
   inet6 addr: fe80::205:4eff:fe51:3c44/64 Scope:Link
   UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
           ...
```

Important fields in this output are the HWaddr (MAC address), inet addr, Bcast (broadcast address) and Mask (network mask).

Routing Information

To view your routing table, use:

```
[matt@pitt:~] route -n
or
[matt@pitt:~] netstat -rn
Kernel IP routing table
Destination  Gateway   Genmask    Flags Metric Ref  Use Iface
192.168.2.0  0.0.0.0   255.255.255.0  U   0   0   0 eth0
0.0.0.0    192.168.2.1 0.0.0.0     UG  0   0   0 eth0
```

The 0.0.0.0 (or default keyword without the -n option) indicates where packets go if they do not match any other destination. In this example, 192.168.2.1 is the gateway device for getting packets out of the local network.

On Red Hat systems, the gateway device is configured as a setting in one of the /etc/sysconfig/network-scripts/ifcfg-<device> files or in /etc/sysconfig/network. On Debian-based systems, this information is also stored in /etc/network/interfaces.

To change the default route from 192.168.2.1 to another network interface, use the commands:

```
[root@pitt:~] route del default
[root@pitt:~] route add default gw 192.168.2.2
[root@pitt:~] route -n
Kernel IP routing table
Destination  Gateway   Genmask    Flags MSS Window irtt Iface
192.168.2.0  0.0.0.0   255.255.255.0  U   0   0   0 eth0
0.0.0.0    192.168.2.2 0.0.0.0     UG  0   0   0 eth0
```

Configure a DHCP Client

To configure DHCP use on a Red Hat system, use the setting 'BOOTPROTO=dhcp' in the /etc/sysconfig/network-scripts/ifcfg-<device> file. On Debian, use a line such as 'iface eth0 inet dhcp' in /etc/network/interfaces.

Common dhcp client programs include dhcpcd (the best), dhclient and pump.

Looking Up Computers by Name

The name of your computer is stored in /etc/HOSTNAME or /etc/hostname (system dependent). It is set with the hostname and dnsdomainname commands at boot time. The hostname and domainname commands are also used to get host and DNS domain name information.

Host name lookups may refer to a number of sources. A local cache of host name to IP addresses are kept in /etc/hosts (networks can be named in /etc/networks but is not used by many programs). The DNS (domain name service) facility can be used as well (and is preferable because of the many computers using the Internet).

Originally, the /etc/host.conf file was used to specify the order of look up for host names. Now, the /etc/nsswitch.conf file is used. The “hosts” entry in nsswitch.conf determines the order:

```
[matt@pitt:~] grep "^hosts:" /etc/nsswitch.conf
hosts:     files dns
```

The output of the above command indicates that /etc/hosts (files) will be checked first, then DNS.

In order for DNS queries to work, the /etc/resolv.conf file must be properly configured. If DHCP is used, this information should be provided automatically. The /etc/resolv.conf file will resemble the following:

```
[matt@pitt:~] cat /etc/resolv.conf
search starnix.com
nameserver 192.168.2.1
nameserver 192.168.2.2
```

The “search” line means that any query that fails will be retried with the .starnix.com string added to the host name. Also, two DNS servers will be queried, in order, for a response.

The commands host, dig and nslookup can be used to test your host name lookup configuration:

```
[matt@pitt:~] host www.linux.com
www.linux.com has address 66.35.250.177
www.linux.com mail is handled by 10 mail.osdn.com.
```

Use the “-t” option to restrict the query to certain DNS record types.

1.112.4 Configure Linux as a PPP client (Weight: 3)

Candidates should understand the basics of the PPP protocol and be able to configure and use PPP for outbound connections.

PPP stands for Point-to-Point Protocol and is a method of providing IP-based networking across devices (such as modems) which do not inherently support IP.

The chat Program

The pppd daemon must be run on both sides. When it is used with modems, a dialer program must be provided. A typical program used for dialing is chat(8). A sample chat script is a set of send-expect pairs and looks like:

```
[matt@pitt:~] chat "" ATZ OK ATDT4165551212 CONNECT "" ogin: user ssword: pass
```

Read in sequence, this means:

- "" - send an empty string to wake up the modem.
- ATZ OK - send an ATZ command to the modem to reset it. Expect the OK response.
- ATDT5551212 CONNECT - dial the modem and wait for a CONNECT string back in response.
- "" ogin: - send nothing but wait for the string 'ogin:' (The 'l' in login is often capitalized. This script avoids the issue).
- user ssword: - send the user name (substitute your remote user name for the string “user), then expect the 'ssword:' string in response.
- pass - send the password.

Running the pppd Daemon

The pppd daemon can be started on the command line with options such as:

```
[root@pitt:~] pppd connect 'chat -f /etc/pppd/chatscript' /dev/ttyS0 57600 crtscts defaultroute
```

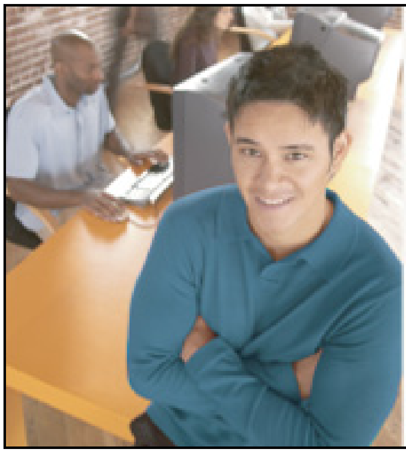
to specify /dev/ttyS0 as the physical device with a baud rate of 57600. Also, we define the program and arguments for making a connection, that we want hardware flow control and a default route added to the system routing table.

Helper Programs

wvdial(1) is an alternative tool to the chat program and is much more intelligent.

wvdialconf(1) can help build a configuration file for use with wvdial. The default location is /etc/wvdial.conf

The /etc/ppp/ip-up and /etc/ppp/ip-down scripts are run by pppd after the connection is brought up or down. These scripts are used to reconfigure routes, interfaces or anything that is required.



Topic 113: Networking Services

1.113.1 Configure and manage xinetd, inetd and related services (Weight: 4)

Configure and manage xinetd, inetd and related services.

Configuration for inetd is in the /etc/inetd.conf file. Configuration for xinetd is in the /etc/xinetd.conf file and files in the /etc/xinetd.d directory. These programs are used to listen on ports on behalf of infrequently accessed services. It keeps resource use to a minimal amount until the service is actually required.

The inetd Service

To add a service to inetd, enter a line such as:

```
pop3  stream tcp  nowait root /usr/sbin/tcpd /usr/sbin/ipop3d
```

The fields are:

Field	Description
service name	The name of the service. The port number is determined by a corresponding entry in /etc/services.
socket type	stream, dgram, raw, rdm or seqpacket. stream and dgram are the most commonly used.
protocol	A valid protocol from /etc/protocols. tcp and udp are commonly used.
flags	wait or nowait.
user	The user name to run the command as. The root user is commonly used.
command	The command (with full path) plus any arguments.

The above example is for a pop3 service. The actual command being run by inetd is the /etc/sbin/tcpd program. This is the TCP wrappers program. It inspects the /etc/hosts.allow and /etc/hosts.deny files to determine additional access restrictions.

The xinetd Service

This is the eXtended inetd service. Each service being offered through xinetd gets its own configuration file in /etc/xinetd.d/.

1.113.2 Operate and perform basic configuration of Mail Transfer Agent (MTA) (Weight: 4)

Candidates should be able to operate and perform basic configuration of a MTA. Advanced custom configurations not included.

The particular internals of sendmail are not covered in the exam, but common functionality between MTAs is covered.

Sendmail configuration files are /etc/sendmail.cf (or /etc/mail/sendmail.cf). It is usually built from the /etc/mail/sendmail.mc file with a number of macro files and the m4 macro language.

Many additional configuration files such as /etc/mail/sendmail.cw are available for specifying which domains the MTA will accept incoming e-mail.

Creating Aliases

Add entries to the /etc/aliases file such as:

```
alias:   recipient1, recipient2
```

This addition will cause e-mail to alias@localdomain to be sent to recipient1 and recipient2. The recipients can be local users, fully qualified e-mail addresses and other aliases.

Once updated, use one of the two commands to notify the system of the new aliases:

```
[root@pitt:~] newaliases
```

or

```
[root@pitt:~] sendmail -bi
```

The latter example is sendmail-specific but most sendmail alternatives will provide a newaliases command.

An individual can also “alias” or forward his or her e-mail address by placing an e-mail address in a .forward file in the home directory, as follows:

```
[matt@pitt:~] cat ~/.forward
matt@otherdomain.com
```

The .forward can also be used to pipe the e-mail into filter programs.

Interacting with Sendmail

Sendmail queues its outgoing mail in /var/spool/mqueue. Locally delivered mail is typically stored in /var/spool/mail/<username>.

To view the messages in the queue, use the command:

```
[root@pitt:~] mailq
```

Stop, start or restart the mail service with:

```
[root@pitt:~] /etc/init.d/sendmail stop
[root@pitt:~] /etc/init.d/sendmail start
```

or

```
[root@pitt:~] /etc/init.d/sendmail restart
```

1.113.3 Operate and perform basic configuration of Apache (Weight: 4)

Candidates should be able to operate and perform basic configuration of Apache. Advanced custom configurations not required.

Starting Apache

To start apache, use the supplied apachectl or apache2ctl (for Apache version 2.x) programs or the init scripts, like:

```
[root@pitt:~] apachectl start
[root@pitt:~] apache2ctl start
```

or

```
[root@pitt:~] /etc/init.d/apache start
```

This script will start a main apache process which runs as the root user. It will spawn child processes to handle actual requests. On Red Hat-based systems, the children typically run as the apache user. On Debian-based systems, the children typically run as the www-data user. As with other services, “stop” and “restart” commands also work.

The apache configuration files are in /etc/httpd, /etc/apache or /etc/apache2 directories, depending on the version of apache and the Linux distribution. The main apache configuration file is httpd.conf. Although srm.conf and access.conf files are available, their use has been deprecated.

1.113.4 Properly manage the NFS and SAMBA daemons (Weight: 4)

Candidates should know how to manage the NFS, smb and nmb daemons.

Using Samba Filesystems

There are two main daemons with Samba:

- smbd - the SMB/CIFS file and printer sharing server. Typically, this is used by Windows clients. However, Linux clients can also access resources.
- nmbd - the name resolution and browsing server. This daemon includes all name resolution, browsing and WINS support that a Windows client would need.

The main configuration file is smb.conf or samba.conf, depending on the distribution. Thee configuration directory also varies depending on the distribution. The most common directories used are /etc and /etc/samba. The file is divided into three main sections: Global Settings, Homes and Printers. More than 300 configuration options exist, so many people use SWAT (Samba Web Admin Tool) for configuration. Other configuration files include /etc/samba/smbpasswd and /etc/samba/smbusers.

1.113.5 Setup and configure basic DNS services (Weight: 4)

Candidates should be able to configure basic DNS services.

DNS server configuration is spread across multiple files:

File	Description
/etc/named.conf	The main configuration file for the server. May be in a directory other than /etc.
zone files	These files define the name to IP address mappings for a given domain name.
reverse zone files	These files define the IP address to name mappings.
hints files	These files provide root server information. The root servers can be used to perform queries if the local files or upstream DNS servers cannot determine an answer.

1.113.7 Set up secure shell (OpenSSH) (Weight: 4)

Candidates should be able to obtain and configure OpenSSH.

A number of programs are related to the secure shell. They include:

Program	Description
sshd	The ssh daemon
ssh	The ssh client program used for remote logins and running remote commands similar to the rsh and telnet programs
scp	A secure copy program similar to rcp
ssh-agent	An end-user program that will securely cache a user's authentication and provide it upon request
ssh-add	The means of loading a user's key(s) into an ssh-agent process
ssh-keygen	A program used to create private/public keys for use with ssh.

Topic 114: Security

1.114.1 Perform security administration tasks (Weight: 4)

Candidates should know how to review system configuration to ensure host security in accordance with local security policies.

Configure TCP wrappers

Use the /etc/hosts.allow and /etc/hosts.deny files. To allow a certain domain in (and all subdomains and machines) but deny everyone else, use a command such as:

```
[matt@pitt:~] cat /etc/hosts.allow
ALL: .foobar.edu
[matt@pitt:~] cat /etc/hosts.deny
ALL: ALL
```

Finding Files that are SUID or SGID

To find any files on your system with any special bits set (SUID, SGID or the sticky bit), use the command:

```
[root@pitt:~] find / -perm +7000
```

Using nmap and netstat Commands

The nmap command is useful for discovering open ports on a remote system. A command such as:

```
[root@pitt:~] nmap localhost
Interesting ports on localhost (127.0.0.1):
PORT   STATE SERVICE
631/tcp open ipp
1665/tcp open netview-aix-5
Nmap finished: 1 IP address (1 host up) scanned in 0.245 seconds
```

The nmap command has many features including OS detection, scan ordering and techniques, firewall evasion and spoofing and much more.

The netstat command is useful if you have login access to the system. The following command lists all open TCP ports on the system:

```
[root@pitt:~] netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address    Foreign Address   State
tcp    0   0 *:netview-aix-5   *:*        LISTEN
tcp    0   0 localhost:ipp    *:*        LISTEN
```

Configuring Firewalling with iptables

The iptables system is for IP packet filtering, inspection and NAT (Network

Address Translation).

Several tables may be defined; each table contains built-in chains plus optional user-defined chains. A chain is a list of rules which have criteria for matching a packet and a target.

The main targets are:

Target	Description
ACCEPT	let the packet through
DROP	drop the packet on the floor
QUEUE	pass the packet to userspace
RETURN	stop traversing the current change and return to the previous chain

1.114.2 Setup host security (Weight: 3)

Candidates should know how to set up a basic level of host security.

Things to be considered when hardening a system:

- Turn off services not required, including services in /etc/inittab, /etc/rcN.d/, /etc/xinetd.conf (or /etc/xinetd.d/), and /etc/inetd.conf.
- Adjust password aging with the passwd or chage commands.
- Set up shadow password. The pwconv, pwunconv, grpconv and grpunconv commands will perform the conversion.
- Forward the root account’s e-mail to another system.
- Make use of the /etc/nologin file when performing sensitive tasks. This file will prevent non-root users from logging in to the system.
- Possibly configure syslogd to log important security-related messages to another syslogd service on a separate host.

1.114.3 Setup user level security (Weight: 1)

Candidates should be able to configure user-level security. Tasks include limits on user logins, processes and memory usage.

Use the ulimit(1) command to list or set user limits. To get a list of current user limits, enter the following:

```
[matt@pitt:~] ulimit -a
core file size     (blocks, -c) 0
data seg size      (kbytes, -d) unlimited
max nice           (-e) 20
```

```
file size        (blocks, -f) unlimited
...
```

A 0 block size for core files is useful because core files leave a trace of what a user's process was doing. The 0 size means that users will not generate a core file on errors in programs.

To set a limit (for debugging) on the core file size, use:

```
[matt@pitt:~] ulimit -c 5000
```

The ulimit command is documented in the bash man page because it is a built-in command.