Глава 7

Управление пакетами

ЦЕЛИ СЕРТИФИКАЦИИ

7.01 Менеджер Пакетов Red Hat
7.02 больше команд RPM
7.03 зависимости и команда yum
7.04 Дополнительные Инструменты Управления Пакетами
Две Минуты ✓ Сверла
Самотестирование Q&A

После завершения установки системы защищены, файловые системы настроены, а другие начальные задачи установки завершены, вам все еще нужно поработать. Почти наверняка, прежде чем ваша система будет в желаемом состоянии, вам потребуется установить или удалить пакеты. Чтобы убедиться, что установлены правильные обновления, вам нужно знать, как заставить систему работать с **Red Hat Subscription Management** (**RHSM**) или репозиториями, связанные с перестроенным дистрибутивом.

Чтобы выполнить эти задачи, вам необходимо понять, как подробно использовать команды **rpm** и **yum**. Хотя это «всего лишь» две команды, они богаты в деталях. Целые книги были посвящены команде **rpm**, такие как Руководство по **RPM** для Red Hat Эрика Фостер-Джонсон. Для многих эта степень глубокого знания команды rpm не требуется, учитывая возможности команды yum и дополнительного пакета инструменты управления, представленные в RHEL 7.

ЦЕЛЬ СЕРТИФИКАЦИИ 7.01
Менеджер пакетов Red Hat

Одной из основных обязанностей системного администратора является управление программным обеспечением. Новые приложения установлены. Сервисы обновляются. Ядро пропатчины. Без правильного инструмента, может быть трудно определить, какое программное обеспечение находится в системе, какое последнее обновление, и какие приложения зависят от другого программного обеспечения. Хуже того, вы можете установить новое программное обеспечение, чтобы потом обнаружить, что это перезаписало важный файл от установленного в настоящее время пакета.

ВНУТРИ экзамена
Административные навыки

Поскольку управление пакетами **RPM** является фундаментальный навык для администраторов Red Hat, разумно ожидать использовать **rpm**, **yum**, и связанные команды на экзамене **RHCSA**. На самом деле, экзамен **RHCE** предполагает эффективное знание таких команд и многое другое, как обязательные навыки. Цели **RHCSA** включают два конкретных требования, рассматриваемые в эта глава:

■ Установка и обновление пакетов программного обеспечения из Red Hat Network, удаленного хранилища или из локальной файловой системы.
■ Обновление пакета ядра соответственно, чтобы обеспечить загрузку системы.
Другая тесно связанная цель - это утилита архивирования **tar**, которая рассматривается в главе 9.

Прежде чем **Red Hat** представила **RPM**-пакеты, **tar** архивы были стандартным методом для распространения программного обеспечения.

Теперь давайте немного разберем эти навыки. Если у вас нет доступа к RHN, не бойтесь. Для RHEL 7, RHN-hosted сервис был свернут в пользу Red Hat Subscription Management (RHSM, которая можно получить доступ через веб-интерфейс из Клиентский портал Red Hat). Вы можете использовать **yum** для того, что бы установить и обновить пакеты из RHSM; вы можно использовать те же команды **yum** для установки и обновления пакетов из стороннего репозитория.

Red Hat Package Manager (RPM) был разработан для решения этих проблем. С RPM, программное обеспечение управляется отдельными пакетами. Пакет RPM включает в себя программное обеспечение с инструкции по добавлению, удалению и обновлению этих файлов. При правильном использовании RPM Система может выполнить резервное копирование ключевых файлов конфигурации, прежде чем приступить к обновлению и удалению.

Rpm и команда **rpm** сосредоточены на отдельных пакетах, которые во многих случаи далеки от идеала и поэтому **rpm** был дополнен командой **yum.**
С подключением к репозиторию, например, доступному от RHSM или стороннего производителя. “rebuilds” например **Scientific Linux** или **CentOS7** вы сможете использовать **yum** для удовлетворения зависимостей автоматически.

Что такое пакет?

В общем смысле **пакет RPM** является контейнером файлов. Она включает в себя группу файлов связанные с конкретной программой или приложением, которое обычно содержит двоичные файлы, сценарии установки, а также файлы конфигурации и документации. Она также включает Инструкции о том, как и где эти файлы должны быть установлены и удалены.

Имя пакета RPM обычно состоит из версии, выпуска и архитектуры для которого он и был построен. Например, вымышленный penguin-3.4.5-26.el7.x86_64.rpm пакет версии 3.4.5, выпуск 26.el7. X86_64 указывает, что он подходит для компьютеров, 64-разрядной архитектуры AMD/Intel.

!!!!
Многие пакеты RPM включают программное обеспечение, скомпилированное для определенного типа ЦП (например, архитектуру x86_64). Тип процессора для системы можно определить с помощью команда uname -i или uname -p. Больше информации о вашем процессоре может находится в /proc/cpuinfo.
!!!!!

Что такое база данных RPM?

В основе этой системы лежит база данных RPM, которая хранится локально на каждой машине в каталоге **/var/lib/rpm**. Помимо прочего, эта база данных отслеживает версию и расположение каждого файла в каждой RPM. База данных RPM также поддерживает контрольную сумму MD5 каждого файла. С помощью контрольной суммы можно использовать команду rpm -V package для определения изменился ли какой-либо

файл из этого пакета RPM. База данных RPM позволяет легко добавлять, удалять и обновлять отдельные пакеты, так как они настроены куда их положить.

RPM также управляет конфликтами между пакетами. Например, предположим, что у вас есть пакет это устанавливает файл конфигурации, и вы хотите обновить от более старой к более новой версии программного обеспечения. Вызовите исходный файл конфигурации /etc/someconfig.conf. Вы уже установленный пакет X. Если затем вы попытаетесь установить более свежую версию пакета X, RPM может быть сконфигурирован для сохранения исходного файла конфигурации и установки нового, как файл /etc/someconfig.conf.rpmnew.

Кроме того, создатель RPM может построить RPM таким образом, что он будет создавать резервные копии оригинал /etc/someconfig.conf (с именем файла, например /etc/someconfig.conf.rpmsave) перед обновлением пакета X заменит файл конфигурации новой версией.

Это может произойти, если формат старого файла конфигурации несовместим с новым версии программного обеспечения.

Хотя обновления RPM должны сохранять или сохранять существующую конфигурацию файлы, нет никаких гарантий, особенно если RPM создан кем-то кроме RedHat. Рекомендуется создать резервную копию всех применимых файлов конфигурации обновление всех связанных пакетов.

Что такое репозиторий?

Пакеты RPM часто организованы в репозитории. Как правило, такие хранилища включает группы пакетов с различными функциями. Например, портал Red Hat дает доступ к следующим репозиториям сервера RHEL 7 (доступны дополнительные репозитории):

- **Red Hat Enterprise Linux Server** основной репозиторий, включающий в себя пакеты, связанные с исходной установкой RHEL 7, а также обновления.
- **RHEL Server Optional** большая группа пакетов с открытым исходным кодом, поддержка от Red Hat.
- **RHEL Server Supplementary(дополнительный)** набор пакетов, выпущенных по лицензиям кроме открытого исходного кода, такого как IBM Java Runtime and Development Kit.
- **RHEL Extras** включает Docker - платформу для упаковки и управления приложениями использование упрощенной формы виртуализации, известной как контейнеры Linux.
- **RHN Tools Client tools** для подписки на **Red Hat Network** через спутниковый сервер, наряду с утилитами для автоматизации Kikstar.

Напротив, категории репозитория для сторонних клонов Red Hat различаются. Как правило, они включать категории, такие, как основной и дополнительные. В большинстве случаев, тогда как основной репозиторий включает только пакеты, доступные с выпущенного DVD, обновленные пакеты настроены в собственном хранилище.

Каждый репозиторий содержит базу данных пакетов в подкаталоге **repodata/**. База данных содержит информацию о каждом пакете и позволяет включать запросы на установку всех зависимостей. Если у вас есть подписка на RHSM, получите доступ к репозиториям Red Hat включен в идентификаторе в файлах **product-id.conf and subscription-manager.conf,** в деректории **/etc/yum/pluginconf.d**. Эти файлы рассматриваются далее в этой главе.

Далее в этой главе вы узнаете, как настроить подключения к репозиториям с помощью файлов конфигурации, связанные с командой **yum.**

!!!!!
Зависимость - это пакет, который необходимо установить, чтобы убедиться, что все функции целевого пакета работают должным образом.
!!!!!

Установите пакет RPM

Есть три основные команды, которые могут установить RPM. Они не будут работать, если есть не установленные зависимости. Например, если вы не установили пакет средства разработки политики SELinux (policycoreutilis-devel) и попытались установить графический интерфейс конфигурации SELinux (policycoreutilis-gui), вы получите следующее сообщение (номера версий могут отличаться):

 # rpm -i policycoreutils-gui-2.2.5-11.el7.x86_64.rpm
error: Failed dependencies:
policycoreutils-devel = 2.2.5-11.el7 is needed by policycoreutils-gui-2.2.5-11.el7.x86_64

Один из способов проверить это - смонтировать RHEL 7 DVD с помощью команды **mount /dev/cdrom /media**. Затем найдите указанный пакет **policycoreutils-gui** в подкаталоге **Packages/**. В качестве альтернативы вы можете загрузить этот пакет непосредственно с **портала Red Hat** или из настроенного репозитория с помощью команды **yumdownloader policycoreutils-gui**. Эта и другие команды **yum** обсуждаются далее в этой главе. Имейте в виду, что некоторые среды Linux с графическим интерфейсом пользователя автоматически монтируют носители **CD/DVD**, которые вставляются в соответствующий дисковод. Если это так, вы увидите каталог **mount** в выходных данных команды **mount**.

Когда отображаются сообщения о зависимостях, **rpm** не устанавливает данный пакет. Обратите внимание на сообщения о зависимостях: для **policycoreutils-gui** требуется пакет **policycoreutils-devel** с тем же номером версии.

!!!!!
Конечно, вы можете использовать опцию --nodeps, чтобы rpm игнорировал зависимости, но это может привести к другим проблемам, если вы не установите эти зависимости как можно скорее. Лучший вариант - использовать соответствующую команду yum, описанную далее в этой главе. В этом случае команда yum install policycoreutils-gui автоматически установит и другие зависимые RPM.
!!!!!

Если вы не остановлены зависимостями, следующие три основные команды могут установить пакеты RPM:

 # rpm -i packagename
 # rpm -U packagename
 # # rpm -F имя пакета

Опция **rpm -i** устанавливает пакет, если он еще не установлен. Опция **rpm -U** обновляет любой существующий пакет или устанавливает его, если более ранняя версия еще не установлена. Опция **rpm -F** обновляет только существующие пакеты. Пакет не устанавливается, если он не был установлен ранее.

Нам нравится добавлять опции **-vh** с помощью команды **rpm**. Эти параметры добавляют подробный режим и используют хэш-метки, которые помогают отслеживать ход установки. Поэтому, когда мы используем **rpm** для установки пакета, мы запускаем следующую команду:

 # rpm -ivh *packagename-version.arch*.rpm

Есть еще одна вещь, связанная с правильно разработанным пакетом **RPM**. При распаковке пакета команда **rpm** проверяет, не перезапишет ли она какие-либо файлы конфигурации. Команда

rpm пытается принимать разумные решения о том, что делать в этой ситуации. Как предлагалось ранее, если команда **rpm** выбирает замену существующего файла конфигурации, она выдает предупреждение (в большинстве случаев), подобное этому:
 # rpm -U penguin-3.26.x86_64.rpm
warning: /etc/someconfig.conf saved as /etc/someconfig.conf.rpmsave

Команда rpm обычно работает таким же образом, когда пакет стирается с **ключом -e**. Если файл конфигурации был изменен, он сохраняется с расширением **.rpmsave** в том же каталоге.

Вам необходимо просмотреть оба файла и определить, какие изменения, если таковые имеются, необходимо внести. Конечно, поскольку не каждый **RPM**-пакет идеален, всегда существует риск, что такое обновление перезапишет этот критически настроенный файл конфигурации. В этом случае резервные копии очень важны.

Обычно команды **rpm** для обновления пакета работают, только если устанавливаемый пакет имеет более новую версию. Иногда желательна более старая версия пакета. Пока со старым пакетом нет проблем с безопасностью, администраторам может быть удобнее работать с более старыми версиями. Ошибки, которые могут быть проблемой в более новом пакете, могут не существовать в более старой версии этого пакета. Поэтому, если вы хотите «понизить» пакет с помощью команды **rpm -i, -U или -F, может помочь ключ --force.**

!!!!!!
Если вы уже настроили пакет, а затем обновили его с помощью *rpm*, проверьте, существует ли сохраненный файл конфигурации, заканчивающийся расширением *.rpmnew*. Используйте его в качестве руководства для изменения настроек в новом файле конфигурации. Но помните, что с обновлениями могут быть дополнительные необходимые изменения. Поэтому вы должны проверить результат для каждой мыслимой производственной среды.
!!!!!!

Удалить пакет RPM

Команда **rpm -e** удаляет пакет. Но сначала **RPM** проверяет несколько вещей. Он выполняет проверку зависимостей, чтобы убедиться, что другим пакетам не нужно то, что вы пытаетесь удалить. Если найдены зависимые пакеты, **rpm -e** завершается ошибкой с сообщением об ошибке, идентифицирующим эти пакеты. С правильно настроенными **RPM**, если вы изменили связанные файлы конфигурации, **RPM** создает копию файла, добавляет расширение **.rpmsave** к концу имени файла, а затем стирает оригинал. Затем он может продолжить удаление. Когда процесс завершен, он удаляет пакет из базы данных.

!!!!!
Будьте очень осторожны с тем, какие пакеты вы удаляете из системы. Как и многие другие утилиты Linux, RPM может молча позволить вам выстрелить себе в ногу. Например, если вы удалите пакеты, включающие работающее ядро, это может сделать эту систему непригодной для использования при следующей загрузке.
!!!!!!

Установите RPM из удаленных систем

С помощью системы **RPM** вы даже можете указать расположение пакетов, аналогичное Интернет-адресу, в формате **URL**. Например, если вы хотите применить команду **rpm** к пакету **foo.rpm** в каталоге **/pub FTP-сервера ftp.rpmdownloads.com**, вы можете установить этот пакет с помощью следующей команды:

 # rpm -ivh ftp://ftp.rpmdownloads.com/pub/foo.rpm

Предполагая, что у вас есть сетевое подключение к этому удаленному серверу, эта конкретная команда **rpm** входит на **FTP-сервер** анонимно и загружает файл. К сожалению, попытка использовать подстановочные знаки в имени пакета с этой командой приводит к

сообщению об ошибке, связанному с «файл не найден». Требуется полное имя пакета, что может раздражать.
Если вы установили RHEL 7 с FTP-сервера, как указано в главах 1 и 2, вы можете заменить связанный **URL-адрес** вместе с точным именем пакета. Например, на основе **FTP-сервера**, настроенного в главе 1, и вышеупомянутого пакета **policycoreutils-gui** соответствующей командой будет

 # rpm -ivh ftp://192.168.122.1/pub/inst/policycoreutils-gui -2.2.5-11.el7.x86_64.rpm

Если для **FTP-сервера** требуются **имя пользователя и пароль**, вы можете включить их в следующем формате

ftp://username:password@hostname:port/path/to/remote/package.rpm

где **username and password** - это имя пользователя и пароль, необходимые для входа в систему, а **порт**, если требуется, указывает **нестандартный порт**, используемый на удаленном сервере **FTP**. На основании предыдущего примера, если имя пользователя - **mjang**, а пароль - **Ila451MS**, вы можете установить **RPM** непосредственно с сервера с помощью следующей команды:

 # rpm -ivh ftp://mjang:Ila451MS@192.168.122.1/pub/inst/policycoreutils-gui -2.2.5- \ 11.el7.x86_64.rpm

Безопасность установки

Безопасность может быть проблемой, особенно с пакетами RPM, загруженными через Интернет. Если хакер «черной шляпы» каким-то образом проникнет в сторонний репозиторий, как вы узнаете, что пакеты из этих источников были подлинными? Ключом является **GNU Privacy Guard (GPG),** который представляет собой реализацию **Pretty Good Privacy (PGP)** с открытым исходным кодом. Если файл **RPM** подписан с использованием закрытого ключа **GPG**, целостность пакета можно проверить с помощью соответствующего открытого ключа **GPG**. Действительная подпись также гарантирует, что пакет был подписан уполномоченной стороной и не был получен от злонамеренного хакера.
Если вы не импортировали или не установили открытые ключи **GPG Red Hat**, вы могли заметить что-то похожее на следующее сообщение при установке пакетов:

warning: vsftpd-3.0.2-9.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID fd431d51: NOKEY

Если вы беспокоитесь о безопасности, это предупреждение должно поднять тревогу. В процессе установки **RHEL 7 ключи GPG** хранятся в каталоге **/etc/pki/rpm-gpg**. Посмотрите на содержимое этого каталога. Вы найдете такие файлы, **как RPM-GPG-KEY-redhat-release**. Чтобы фактически использовать ключ для проверки пакетов, его нужно импортировать, а команда для импорта ключа **GPG** довольно проста:

 # rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Если выходных данных нет, команда **rpm**, вероятно, успешно импортировала ключ **GPG**. Даже если эта команда будет выполнена успешно, если вы повторите ее, появится сообщение «Ошибка импорта». Кроме того, ключ **GPG** теперь включен в базу данных **RPM**, что можно проверить с помощью команды **rpm -qa gpg-pubkey**.
В каталоге **/etc/pki/rpm-gpg** обычно доступно пять ключей GPG, как описано в **таблице 7-1**.
Далее в этой главе вы увидите, как ключи **GPG** автоматически импортируются из удаленных репозиториев при установке новых пакетов.

ТАБЛИЦА 7-1 Ключи GPG для проверки обновлений программного обеспечения

Ключ GPG	Описание
RPM-GPG-KEY-redhat-beta	Пакеты, созданные для бета-версии RHEL 7
RPM-GPG-KEY-redhat-legacy-former	Пакеты для выпусков до января 2007 (и обновления)
RPM-GPG-KEY-redhat-legacy-release	Пакеты для выпусков после января 2007 года
RPM-GPG-KEY-redhat-legacy-rhx	Пакеты, связанные с Red Hat Exchange
RPM-GPG-KEY-redhat-release	Выпущенные пакеты для RHEL 7

Специальные процедуры RPM с ядром

Обновленные ядра включают новые функции, решают проблемы безопасности и в целом помогают системам **Linux** работать лучше. Тем не менее, обновления ядра могут работать неправильно и препятствовать загрузке системы или вызывать сбой приложений; это особенно распространено, если были установлены специализированные пакеты, которые зависят от существующей версии ядра.

Если вам известно о любом программном обеспечении, использующем пользовательский модуль ядра, не обновляйте ядро, если вы не готовы повторить каждый шаг по настройке программного обеспечения с использованием существующего ядра, будь то получение новых модулей ядра с закрытым исходным кодом. от производителя для новой версии, перестройка специализированных модулей для нового ядра или другая ручная работа. Например, драйверы для нескольких беспроводных сетевых карт и принтеров без встроенных в дерево драйверов с открытым исходным кодом могут быть привязаны к конкретной версии ядра. Некоторые программные компоненты виртуальной машины (не включая **KVM**) могут быть установлены для конкретной версии ядра.

Если вы видите доступное обновление для **RPM** ядра, возникает соблазн запустить команду **rpm -U newkernel**. Не делай этого! Он перезаписывает ваше существующее ядро, и если обновленное ядро не работает с системой, вам не повезло. (Ну, не совсем не повезло, но если вы перезагрузитесь и у вас возникнут проблемы, вам придется использовать режим восстановления, описанный в главе 5, для загрузки системы и переустановки существующего ядра. В те дни, когда проводились отдельные Разделы «Техническое обслуживание системы» на экзаменах Red Hat, которые могли бы быть использованы для интересного тестового сценария.) Лучший вариант для обновления до нового ядра - это установить его, в частности, с помощью такой команды:

 # rpm -ivh *newkernel*

Если вы подключены к соответствующему хранилищу, следующая команда работает одинаково хорошо:

 # yum install kernel

Это устанавливает новое ядро вместе со связанными файлами рядом с текущим рабочим ядром. Один пример результата показан на рисунке 7-1 в выводе команды **ls /boot**.

!!!!!
Также можно безопасно установить новое ядро, запустив *yum update kernel*. Фактически, по умолчанию, yum настроен так, чтобы всегда устанавливать пакет ядра и оставлять любое старое ядро на месте. Это относится максимум к трем ядрам, установленным одновременно.
!!!!

В **таблице 7-2** кратко описаны различные файлы для различных частей процесса загрузки в каталоге **/boot.**

РИСУНОК 7-1 Новые и существующие файлы ядра в /boot каталоге

```
[root@server1 ~]# ls /boot
config-3.10.0-123.13.2.el7.x86_64
config-3.10.0-123.el7.x86_64
grub2
initramfs-0-rescue-b37be8dd26f97ac4ba4a6152f5e92b44.img
initramfs-3.10.0-123.13.2.el7.x86_64.img
initramfs-3.10.0-123.el7.x86_64.img
initrd-plymouth.img
symvers-3.10.0-123.13.2.el7.x86_64.gz
symvers-3.10.0-123.el7.x86_64.gz
System.map-3.10.0-123.13.2.el7.x86_64
System.map-3.10.0-123.el7.x86_64
vmlinuz-0-rescue-b37be8dd26f97ac4ba4a6152f5e92b44
vmlinuz-3.10.0-123.13.2.el7.x86_64
vmlinuz-3.10.0-123.el7.x86_64
[root@server1 ~]#
```

Установка нового ядра добавляет параметры для загрузки нового ядра в файле конфигурации **GRUB (/boot/grub2/grub.cfg),** не стирая существующие параметры. Сокращенная версия пересмотренного файла конфигурации **GRUB** показана на **рисунке 7-2.**

Внимательное прочтение двух разделов «**menuentry**» показывает, что единственное различие заключается в номерах версий, указанных в заголовке для ядра **Linux** и исходной файловой системы RAM-диска. По умолчанию система будет загружаться с новым установленным ядром. Поэтому, если это ядро не работает, вы можете перезагрузить систему, получить доступ к меню **GRUB**, а затем загрузиться со старого, ранее работавшего ядра.

Таблица 7-2 Файлы в каталоге /boot

Файл	Описание
config-*	Параметры конфигурации ядра; текстовый файл
grub2/	Каталог с файлами конфигурации GRUB
initramfs-*	Начальная файловая система RAMdisk, корневая файловая система, вызываемая в процессе загрузки для загрузки других компонентов ядра, таких как модули блочных устройств
initrd-plymouth.img	RAM дисковая файловая система, содержащая файлы для графической анимации, отображаемой при загрузке Plymouth
symvers-*	Список модулей
System.map-*	Карта системных имен переменных и функций с их расположением в памяти
vmlinuz-*	Фактическое ядро Linux

РИСУНОК 7-2 GRUB со вторым ядром из файла /boot/grub2/grub.cfg

```
menuentry 'Red Hat Enterprise Linux Server (3.10.0-123.13.2.el7.x86_64) 7.0 (Maipo)' --class
red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.
10.0-123.el7.x86_64-advanced-d055418f-1ff6-46bf-8476-b391e82a6f51' {
        # output removed for brevity
        set root='hd0,msdos1'
        linux16 /vmlinuz-3.10.0-123.13.2.el7.x86_64 root=/dev/mapper/rhel-root ro rd.lvm.lv=r
hel/root vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel/swap crashkernel=auto  vconsole.keyma
p=uk rhgb quiet LANG=en_GB.UTF-8
        initrd16 /initramfs-3.10.0-123.13.2.el7.x86_64.img
}
menuentry 'Red Hat Enterprise Linux Server (3.10.0-123.el7.x86_64) 7.0 (Maipo)' --class red -
-class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-
123.el7.x86_64-advanced-d055418f-1ff6-46bf-8476-b391e82a6f51' {
        # output removed for brevity
        set root='hd0,msdos1'
        linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=/dev/mapper/rhel-root ro rd.lvm.lv=rhel/r
oot vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel/swap crashkernel=auto  vconsole.keymap=uk
rhgb quiet LANG=en_GB.UTF-8
        initrd16 /initramfs-3.10.0-123.el7.x86_64.img
}
```

ЦЕЛЬ СЕРТИФИКАЦИИ 7.02

Больше RPM команд

Команда **rpm** богата деталями. Все, что может сделать эта книга, - это рассказать о некоторых основных способах, которыми **rpm** может помочь вам управлять **RHEL**. Вы уже читали о том, как **rpm** может устанавливать и обновлять пакеты различными способами. Запросы(**Queries**) помогут вам подробно определить, что установлено. Инструменты проверки могут помочь вам проверить целостность пакетов и отдельных файлов. Вы можете использовать связанные инструменты, чтобы определить назначение различных **RPM**, а также узнать полный список уже установленных **RPM**.

Пакетные запросы(Queries)

Самый простой запрос RPM проверяет, установлен ли конкретный пакет. Следующая команда проверяет установку пакета systemd (номер версии может отличаться):
 # rpm -q systemd
Systemd-208-11.el7.x86_64

Вы можете сделать больше с запросами **RPM**, как описано в **Таблице 7-3**. Обратите внимание, как запросы связаны с **-q или --query**; переключатели полного слова, такие как **--query**, обычно ассоциируются, начинается с двойной черты.

ТАБЛИЦА 7-3 rpm --query Options

Команда запроса rpm	Значение
rpm -qa	Перечисляет все установленные пакеты.
rpm -qf /path/to/file	Определяет пакет, связанный с /Путь/к/файлу.
rpm -qc имя_пакета	Перечисляет только файлы конфигурации из packagename.
rpm -qd имя_пакета	Перечисляет только файлы документации от packagename.
rpm -qi packagename	Отображает основную информацию для packagename.
rpm -ql имя_пакета	Перечисляет все файлы из packagename.
rpm -qR имя_пакета	Отмечает все зависимости; Вы не можете установить packagename без них.

rpm -q --changelog packagename	Отображает информацию об изменении для packagename.

Если вы хотите запросить файл пакета RPM, а не локальную базу данных RPM, все вам нужно добавить ключ -p и указать путь или URL файла пакета.
Например, следующая команда перечисляет все файлы пакета RPM EPEL-релиз-7-5.noarch.rpm:

 # rpm -qlp epel-release-7-5.noarch.rpm

Цифровая подпись пакета

RPM использует несколько методов для проверки целостности пакета. Вы видели, как импортировать ключ **GPG**. Некоторые из доступных методов отображаются при проверке пакета с помощью команды **rpm --checksig pkg.rpm**. (**Ключ -K** эквивалентен параметру **--checksig**.) Например, если вы загрузили пакет от третьей стороны, например, гипотетический пакет **pkg-1.2.3-4.noarch.rpm**, и хотите проверить его на соответствие импортированные ключи **GPG**, выполните следующую команду:

 # rpm --checksig pkg-1.2.3-4.noarch.rpm

В случае успеха вы увидите вывод, подобный следующему:

pkg-1.2.3-4.noarch.rpm: rsa sha1 (md5) pgp md5 OK

Это гарантирует, что пакет является подлинным, а **RPM**-файл не был изменен третьей стороной. Вы уже можете распознать алгоритмы, используемые для проверки целостности пакета:

- **rsa** Названный по имени его создателей, Rivest, Shamir и Adleman, это алгоритм шифрования с открытым ключом.
- **sha1** 160-битный дайджест сообщения Secure Hash Algorithm; криптографическая хеш-функция.
- **md5** Message Digest 5, криптографическая хеш-функция.
- **pgp** PGP, как это реализовано в Linux компанией GPG.

Проверка файла

Проверка установленного пакета сравнивает информацию об этом пакете с информацией из базы данных **RPM** в системе. Ключ **--verify (или -v)** проверяет размер, контрольную сумму MD5, разрешения, тип, владельца и группу каждого файла в пакете. Проверка может быть выполнена несколькими способами. Вот несколько примеров:

- Проверить все файлы. Естественно, это может занять много времени в вашей системе. (Конечно, команда **rpm -Va** выполняет ту же функцию.)

  # rpm --verify –a

- Проверьте все файлы в пакете по загруженному RPM.

  # rpm --verify -p vsftpd-3.0.2-9.el7.x86_64.rpm

- yuПроверьте файл, связанный с конкретным пакетом.

 # rpm --verify --file /bin/ls

 # rpm --verify --file /bin/vi
.....G. /bin/vi

В таблице 7-4 перечислены коды ошибок и их значения.

ТАБЛИЦА 7-4 rpm --verify Codes

Код ошибки	**Имеется в виду**
5	Контрольная сумма MD5
S	Размер файла
L	Символьная ссылка
T	Время модификации файла
D	устройство
U	пользователь
G	группа
M	Режим(mode)

Теперь вот интересный эксперимент: если у вас установлена одна версия пакета, используйте команду **rpm --verify -p** со второй версией того же пакета. Нахождение таких пакет не должен быть слишком сложным, потому что **Red Hat** обновляет пакеты для обновлений функций, исправлений безопасности и, да, исправлений ошибок. Например, когда мы писали эту книгу для RHEL 7, у нас был доступ к **sssd-client-1.11.2-65.el7.x86_64.rpm и sssd-client-1.11.2-28.el7.x86_64.rpm**. Когда последняя версия была установлена, мы запустили команду

 # rpm --verify -p sssd-client-1.11.2-65.el7.x86_64.rpm

и получил полный список измененных файлов, как показано на рисунке 7-3. Эта команда предоставляет информацию о том, что было изменено между различными версиями пакета **sssd-client**.

РИСУНОК 7-3 Проверка изменений между пакетами

```
[root@server1 Packages]# rpm --verify -p sssd-client-1.11.2-65.el7.x86_64.rpm
S.5....T.    /usr/lib64/krb5/plugins/authdata/sssd_pac_plugin.so
S.5....T.    /usr/lib64/krb5/plugins/libkrb5/sssd_krb5_locator_plugin.so
S.5....T.    /usr/lib64/libnss_sss.so.2
S.5....T.    /usr/lib64/security/pam_sss.so
missing     /usr/share/doc/sssd-client-1.11.2
missing   d /usr/share/doc/sssd-client-1.11.2/COPYING
missing   d /usr/share/doc/sssd-client-1.11.2/COPYING.LESSER
missing   d /usr/share/man/ca/man8/pam_sss.8.gz
missing   d /usr/share/man/es/man8/pam_sss.8.gz
S.5....T.  d /usr/share/man/es/man8/sssd_krb5_locator_plugin.8.gz
S.5....T.  d /usr/share/man/fr/man8/pam_sss.8.gz
S.5....T.  d /usr/share/man/fr/man8/sssd_krb5_locator_plugin.8.gz
missing   d /usr/share/man/ja/man8/pam_sss.8.gz
S.5....T.  d /usr/share/man/ja/man8/sssd_krb5_locator_plugin.8.gz
S.5....T.  d /usr/share/man/man8/pam_sss.8.gz
S.5....T.  d /usr/share/man/man8/sssd_krb5_locator_plugin.8.gz
S.5....T.  d /usr/share/man/uk/man8/pam_sss.8.gz
S.5....T.  d /usr/share/man/uk/man8/sssd_krb5_locator_plugin.8.gz
[root@server1 Packages]#
```

ЦЕЛЬ СЕРТИФИКАЦИИ 7.03

Зависимости и команда yum

Команда **yum** позволяет легко добавлять и удалять программные пакеты в системе и из нее. Он поддерживает базу данных о том, как правильно добавлять, обновлять и удалять пакеты. Это позволяет относительно просто добавлять и удалять программное обеспечение с помощью одной команды. Эта единственная команда преодолела то, что было известно как «ад зависимости»(“dependency hell.”).

Команда **yum** изначально была разработана для **Yellow Dog Linux**. Название основано на обновленной версии **Yellow Dog**. Учитывая проблемы, связанные с адской зависимостью, пользователи Linux были мотивированы, чтобы найти решение. Он был адаптирован для дистрибутивов **Red Hat** с помощью разработчиков из **Duke University**.

Конфигурация **yum** зависит от библиотек пакетов, известных как **репозитории**. **Репозитории Red Hat** доступны через портал **Red Hat**, в то время как репозитории сторонних дистрибутивов восстановления используют свои общедоступные серверы. В любом случае важно знать, как работает команда **yum**, а также как она устанавливает и обновляет отдельные пакеты и группы пакетов.

Пример ада зависимости (“dependency hell.”).

Чтобы больше узнать о необходимости использования команды **yum**, изучите **рисунок 7-4**. Вам не нужен файл **kernel.spec**. Пакеты, перечисленные на этом рисунке, - это то, что требуется для создания RPM. Хотя создание пакета RPM не является обязательным требованием к экзамену, соответствующие пакеты иллюстрируют необходимость использования **yum**.

Вы можете попробовать использовать команду **rpm** для установки каждого из этих пакетов. Для этого выполните следующие действия:

1. Включите DVD RHEL 7. Вставьте его в дисковод или убедитесь, что он включен в конфигурацию целевой виртуальной машины.
2. Если он не смонтирован, подключите этот DVD с помощью следующей команды. Конечно, другой / пустой каталог может быть заменен на **/media.**
   # mount /dev/cdrom/media

3. Перейдите в каталог, где смонтирован DVD-диск (то есть **/media** или какой-либо подкаталог **/media**).
4. Пакеты RPM на DVD-диске RHEL 7 можно найти в подкаталоге **Packages/** DVD-диска. Перейдите в этот подкаталог.
5. Введите команду **rpm -ivh**, а затем введите имена пакетов, перечисленных на **рис. 7-4**. Для этой цели может быть проще использовать завершение команды; например, если вы должны были ввести
   # rpm -ivh gcc-
   затем вы можете дважды нажать клавишу **Tab** и просмотреть доступные пакеты, начинающиеся с **gcc-**. Затем вы можете ввести дополнительные клавиши и снова нажать клавишу табуляции, чтобы завершить название пакета. После небольшой работы вы получите нечто похожее на команду и результаты, показанные на **рис. 7-5**. То, что на самом деле появляется, зависит от текущего уровня редакции каждого пакета, а также от того, что уже установлено в локальной системе.
6. Следующим шагом будет попытка включить отсутствующие зависимости в список пакетов, которые нужно установить. Когда мы попробуем этот шаг, это приведет к большему количеству зависимостей, как показано на **рисунке 7-6.**

На этом этапе добавление большего количества пакетов к установке становится несколько сложнее. Откуда вы знаете, кроме как из опыта, что пакет mpfr- * будет

удовлетворить сообщение **«Failed Dependencies» для libmpfr.so.4 () (64bit)**? Даже если вы уже поняли, включение таких пакетов не достаточно. Есть даже еще один уровень зависимых пакетов. Эта боль известна как адовы зависимости.

РИСУНОК 7-4 Пакеты, необходимые для сборки RPM

```
[root@server1 SPECS]# rpmbuild -ba kernel.spec
error: Failed build dependencies:
        gcc >= 3.4.2 is needed by kernel-3.10.0-123.13.2.el7.x86_64
        xmlto is needed by kernel-3.10.0-123.13.2.el7.x86_64
        hmaccalc is needed by kernel-3.10.0-123.13.2.el7.x86_64
        elfutils-devel is needed by kernel-3.10.0-123.13.2.el7.x86_64
        binutils-devel is needed by kernel-3.10.0-123.13.2.el7.x86_64
        python-devel is needed by kernel-3.10.0-123.13.2.el7.x86_64
        perl(ExtUtils::Embed) is needed by kernel-3.10.0-123.13.2.el7.x86_64
        bison is needed by kernel-3.10.0-123.13.2.el7.x86_64
        audit-libs-devel is needed by kernel-3.10.0-123.13.2.el7.x86_64
        numactl-devel is needed by kernel-3.10.0-123.13.2.el7.x86_64
[root@server1 SPECS]#
```

РИСУНОК 7-5. Эти пакеты имеют зависимости.

```
[root@server1 Packages]# rpm -ivh gcc-4.8.2-16.el7.x86_64.rpm xmlto-0.0.25-7.el7.x86_64.rpm
hmaccalc-0.9.13-4.el7.x86_64.rpm elfutils-devel-0.158-3.el7.x86_64.rpm binutils-devel-2.23.5
2.0.1-16.el7.x86_64.rpm python-devel-2.7.5-16.el7.x86_64.rpm perl-ExtUtils-Embed-1.30-283.el
7.noarch.rpm bison-2.7-4.el7.x86_64.rpm audit-libs-devel-2.3.3-4.el7.x86_64.rpm numactl-deve
l-2.0.9-2.el7.x86_64.rpm
error: Failed dependencies:
        cpp = 4.8.2-16.el7 is needed by gcc-4.8.2-16.el7.x86_64
        glibc-devel >= 2.2.90-12 is needed by gcc-4.8.2-16.el7.x86_64
        libmpc.so.3()(64bit) is needed by gcc-4.8.2-16.el7.x86_64
        libmpfr.so.4()(64bit) is needed by gcc-4.8.2-16.el7.x86_64
        flex is needed by xmlto-0.0.25-7.el7.x86_64
        elfutils-libelf-devel(x86-64) = 0.158-3.el7 is needed by elfutils-devel-0.158-3.el7.
x86_64
        perl-devel is needed by perl-ExtUtils-Embed-0:1.30-283.el7.noarch
        kernel-headers >= 2.6.29 is needed by audit-libs-devel-2.3.3-4.el7.x86_64
[root@server1 Packages]#
```

РИСУНОК 7-6. Есть еще больше зависимостей.

```
[root@server1 Packages]# rpm -ivh gcc-4.8.2-16.el7.x86_64.rpm xmlto-0.0.25-7.el7.x86_64.rpm
hmaccalc-0.9.13-4.el7.x86_64.rpm elfutils-devel-0.158-3.el7.x86_64.rpm binutils-devel-2.23.5
2.0.1-16.el7.x86_64.rpm python-devel-2.7.5-16.el7.x86_64.rpm perl-ExtUtils-Embed-1.30-283.el
7.noarch.rpm bison-2.7-4.el7.x86_64.rpm audit-libs-devel-2.3.3-4.el7.x86_64.rpm numactl-deve
l-2.0.9-2.el7.x86_64.rpm cpp-4.8.2-16.el7.x86_64.rpm glibc-devel-2.17-55.el7.x86_64.rpm libm
pc-1.0.1-3.el7.x86_64.rpm mpfr-3.1.1-4.el7.x86_64.rpm flex-2.5.37-3.el7.x86_64.rpm elfutils-
libelf-devel-0.158-3.el7.x86_64.rpm perl-devel-5.16.3-283.el7.x86_64.rpm kernel-headers-3.10
.0-123.el7.x86_64.rpm
error: Failed dependencies:
        glibc-headers is needed by glibc-devel-2.17-55.el7.x86_64
        glibc-headers = 2.17-55.el7 is needed by glibc-devel-2.17-55.el7.x86_64
        gdbm-devel is needed by perl-devel-4:5.16.3-283.el7.x86_64
        libdb-devel is needed by perl-devel-4:5.16.3-283.el7.x86_64
        perl(ExtUtils::Installed) is needed by perl-devel-4:5.16.3-283.el7.x86_64
        perl(ExtUtils::MakeMaker) is needed by perl-devel-4:5.16.3-283.el7.x86_64
        perl(ExtUtils::ParseXS) is needed by perl-devel-4:5.16.3-283.el7.x86_64
        systemtap-sdt-devel is needed by perl-devel-4:5.16.3-283.el7.x86_64
[root@server1 Packages]#
```

Освобождение от ада зависимостей

До команды **yum** некоторые попытки использовать команду **rpm** были остановлены ранее описанными зависимостями. Конечно, вы можете установить эти зависимые пакеты с помощью одной и той же команды, но что, если эти зависимости сами имеют зависимости? Это, пожалуй, самое большое преимущество команды yum.

До этого RHEL включал разрешение зависимостей в процесс обновления. Через RHEL 4 это было сделано с помощью **up2date**. Red Hat включает **yum**, начиная с RHEL 5. Команда **yum** использует подписанные каналы **Red Hat Portal** и любые другие **репозитории**, настроенные в каталоге **/etc/yum.repos.d.**

Все, что вам нужно сделать для установки пакетов, перечисленных на **рис. 7-4**, - запустить следующую команду:

**# yum install gcc xmlto hmaccalc elfutils-devel binutils-devel **
> python-devel perl-ExtUtils-Embed bison audit-libs-devel numactl-devel

Если будет предложено, примите запрос на установку дополнительных зависимых пакетов, после чего все отмеченные зависимости будут установлены автоматически. (Да, **ключ -y** будет выполнять ту же функцию.) Если обновления доступны из подключенных хранилищ, последние доступные версия каждого пакета установлена. Команда **yum** более подробно описана ниже в этой главе.

версия каждого пакета установлена. Команда yum более подробно описана ниже в этой главе.

Но если вы используете RHEL 7 без подключения к Red Hat Portal, ничего не произойдет. Вскоре вы увидите, как создать соединение между yum и сервером установки, созданным в главе 1.

Ряд сторонних репозиториев доступны для RHEL. Они включают несколько популярных приложений, которые не поддерживаются Red Hat. Например, один из авторов этой книги использует внешний репозиторий для установки пакетов, связанных с его портативной беспроводной сетевой картой.

Хотя владельцы этих репозиториев тесно сотрудничают с некоторыми разработчиками Red Hat, есть некоторые отчеты, в которых зависимости, требуемые от одного репозитория, недоступны из других репозиториев, что приводит к другой форме «ада зависимостей». Однако, более популярные сторонние репозитории работают отлично; мы никогда не встречали «ад зависимости» при использовании этих репозиториев.

!!!!!!
Существует две основные причины, по которым Red Hat не включает наиболее проверенные и популярные пакеты, доступные в сторонних репозиториях. Некоторые не выпускаются по лицензиям с открытым исходным кодом, а другие являются пакетами, которые Red Hat просто не поддерживает.
!!!!!!!!!

Базовая конфигурация yum

Избавление от адской зависимости зависит от правильной настройки **yum**. Вам нужно не только знать, как настроить **yum** для подключения к репозиториям через Интернет, но также вы должны знать, как настроить **yum** для подключения к репозиториям в локальной сети. Обладая этими знаниями, вы можете подключать yum к репозиториям на Red Hat Portal, к репозиториям, настроенным третьими лицами, и к настраиваемым репозиториям, настроенным для специализированных сетей. И помните, что во время экзаменов в Red Hat у вас не будет доступа к Интернету.

Для этого вам нужно понять, как настраевается **yum** в деталях. Он начинается с файла конфигурации **/etc/yum.conf** и продолжается файлами в **/etc/yum** и **/etc/yum.repos.d**

каталоги. Чтобы получить полный список директив конфигурации **yum** и их текущих значений, выполните следующую команду:

 # yum-config-manager

Эта команда запрашивает установки пакета **yum-utils**.

Базовый файл конфигурации yum: yum.conf

В этом разделе построчно анализируется версия файла **/etc/yum.conf** по умолчанию. Несмотря на то, что вы не будете вносить изменения в этот файл в большинстве случаев, вам нужно понять хотя бы стандартные директивы этого файла, если что-то пойдет не так. Следующие строки являются прямыми выдержками из версии этого файла по умолчанию.

[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3

Первая директива является заголовком; заголовок **[main]** предполагает, что все следующие директивы применяются глобально к **yum**:

[main]

Директива **cachedir** указывает, куда следует загружать кэши пакетов, списков пакетов и связанных баз данных. Основанный на стандартной 64-битной архитектуре для RHEL 7, он преобразуется в каталог **/var/cache/yum/x86_64/7Server**.

cachedir=/var/cache/yum/$basearch/$releasever

Булева директива **keepcache** указывает, действительно ли **yum** хранит загруженные заголовки и пакеты в каталоге, указанном **cachedir**. Показанный здесь стандарт предполагает, что кеши не хранятся, что помогает обеспечить актуальность системы с последними доступными пакетами (за счет немного более медленного выполнения **yum**, так как метаданные выполняются при каждом выполнении).

keepcache = 0

Директива **debuglevel** тесно связана с директивами **errorlevel и logfile**, так как они определяют детали, связанные с сообщениями об отладке и ошибках. Даже если директива **errorlevel** не отображается, по умолчанию она и **debuglevel** установлены на 2. Доступный диапазон: 0–10, где 0 практически не предоставляет информации, а 10 - слишком много, даже для разработчиков.

debuglevel=2

logfile=/var/log/yum.log

Булева директива **surearch** обеспечивает соответствие архитектуры фактическому типу процессора, определенному командой **arch**.

exactarch = 1

Булева директива **obsoletes** может поддерживать удаление устаревших пакетов вместе с командой **yum update**.

obsoletes=1

Логическая директива **gpgcheck** проверяет, действительно ли команда **yum** проверяет подпись **GPG** загруженных пакетов.

gpgcheck = 1

Булева директива **plugins** предоставляет необходимую ссылку на плагины **RHN** на основе **Python** в каталоге **/usr/share/yum-plugins**. Это также косвенно относится к файлам конфигурации плагина в каталоге **/etc/yum/pluginconf.d**.

plagins=1

Директива **installonly_limit** указывает, сколько пакетов, перечисленных в опции **installonlypkgs** (обычно это ядро), может быть установлено одновременно:

installonly_limit=3

Чтобы обеспечить актуальность данных заголовка, загруженных из **RHN** (и любых других репозиториев), директива **metadata_expire** указывает время жизни заголовков. Хотя в комментариях в **yum.conf** указано, что значение по умолчанию составляет **90** минут, фактическое значение по умолчанию для RHEL 7 составляет шесть часов. Другими словами, если вы не использовали команду **yum** в течение последних шести часов, при следующем использовании команды **yum** загружается самая последняя информация заголовка.

 # metadata_expire = 90m

Последняя интересная директива, в комментариях, оказывается по умолчанию; это ссылка на отмеченный каталог для актуальной информации о конфигурации репозиториев:

 # PUT YOUR REPOS HERE OR IN separate files named file.repo
 # in /etc/yum.repos.d

Файлы конфигурации в каталоге /etc/yum/pluginconf.d

Файлы по умолчанию в каталоге **/etc/yum/pluginconf.d** настраивают соединение между **yum** и **Red Hat Portal** или локальным сервером **Satellite**. Если вы учитесь в дистрибутиве перестройки **RHEL**, таком как **CentOS**, вы увидите другой набор файлов в этом каталоге. В **CentOS** файлы в этом каталоге ориентированы на подключение

локальной системы к лучшим репозиториям через Интернет. Однако это книга Red Hat, поэтому основное внимание будет уделено двум основным файлам в установке RHEL 7.

Red Hat Network Plagins (Сетевые Плагины)

Если у вас есть подписка на **RHN** через старую версию **Red Hat Satellite Server**, файл **rhnplugin.conf** в этом каталоге особенно важен. Хотя приведенные ниже директивы могут показаться простыми, они разрешают доступ и проверяют подписи **GPG**:

```
[main]
enabled = 1
gpgcheck = 1
timeout = 120
```

В комментариях этот файл предполагает, что для разных репозиториев можно настроить разные параметры. Репозитории, заключенные в скобки, должны соответствовать тем, которые связаны с реальными репозиториями **RHN**.

Плагины управления подпиской Red Hat

Файлы **subscription-manager.conf** и **product-id.conf** предназначены для подключения системы **yum** к порталу **Red Hat** с помощью диспетчера подписок. Как будет обсуждаться далее в этой главе, **Subscription Manager** - это система, предназначенная для замены **RHN** для обновлений системы. Файл **subscription-manager.conf** очень прост, с двумя директивами, которые обеспечивают соединение между **yum** и плагином **Subscription Manager:**

```
[main]
enabled=1
```

Файлы конфигурации в каталоге /etc/yum.repos.d

Файлы конфигурации в каталоге **/etc/yum.repos.d** предназначены для подключения систем к реальным репозиториям. Если вы используете перестроенный дистрибутив, такой как **CentOS**, вы увидите файлы, которые подключаются к общедоступным репозиториям в Интернете. Если вы используете RHEL 7, этот каталог может быть пустым, если только система не была зарегистрирована в **Red Hat Subscription Management**. В этом случае вы увидите файл **redhat.repo** в этом каталоге, который предназначен для получения дополнительных обновлений от портала **Red Hat**.

Пара общих элементов для файлов конфигурации в каталоге **/etc/yum.repos.d** - это расширение файла **(.repo)** и документация, доступная с помощью команды **man yum.conf**.

Правильно настроенный файл **.repo** в каталоге **/etc/yum.repos.d** может быть очень удобен для установки групп пакетов с помощью команды **yum**. Поскольку каталог **/etc/yum.repos.d** может быть пустым в системе **RHEL 7**, вы должны знать, как создать этот файл с нуля, используя данные для сервера установки и информацию, доступную на справочной странице **yum.conf**.

Понимание /etc/yum.repos.d Файлы конфигурации для перестроенных дистрибутивов(клонов RH)

Если вы используете перестроенные(клоны RH) дистрибутив, файлы в каталоге **/etc/yum.repos.d** могут подключить локальную систему к одному или нескольким удаленным репозиториям. Один пример взят из **CentOS 7**, как показано на **рисунке 7-7**.

Хотя он включает в себя несколько различных репозиториев, вы можете изучить шаблон директив, настроенный для каждого репозитория.

РИСУНОК 7-7. Несколько репозиториев, настроенных в одном файле

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

На **рисунке 7-7** показаны четыре строфы. Каждый раздел представляет собой соединение с **репозиторием CentOS**. Например, первая строфа включает в себя основные элементы хрануилища и многое другое. В первой строке в скобках указано имя хранилища. В этом случае **[base]** просто означает базовый репозиторий, используемый дистрибутивом **CentOS 7**. Он не представляет каталог, в котором установлены соответствующие пакеты.

[basa]

Тем не менее, когда вы запускаете команду **yum update** для обновления локальной базы данных этих удаленных пакетов, она включает в себя **base** в качестве имени репозитория в выводе, аналогичном следующему, что предполагает, что для загрузки базы данных 3.6KB существующей базы данных потребовалась одна секунда данные хранилища:

basa | 3,6 КБ 00:00:01

Хотя имя репозитория следует, оно только для целей документации и не влияет на то, как пакеты или базы данных пакетов читаются или загружаются. Однако включение директивы **name** позволяет избежать нефатального сообщения об ошибке.

name = CentOS- $ releasever – Basa

Обратите внимание на следующую директиву **mirrorlist**. Он указывает **URL**-адрес файла, который содержит список из нескольких **URL**-адресов ближайших удаленных серверов с копией фактического хранилища пакетов. Обычно он работает с протоколом **HTTP или FTP**. (Он может даже работать с локальными каталогами или подключенными общими файловыми системами, как описано в **упражнении 7-1**.)

mirrorlist=http://mirrorlist.centos.org/?release=$releasever &arch=$basearch&repo=os

Кроме того, эти репозитории могут быть настроены в файле, загруженном с помощью директивы **baseurl:**

#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/

Репозитории, настроенные в файлах **.repo** в каталоге **/etc/yum.repos.d**, включены по умолчанию. Следующая директива предоставляет простой способ деактивировать соединение с таким (**enabled=1** активирует соединение):

enabled=0

Если вы хотите отключить подписи **GPG** для каждого загружаемого пакета, следующая команда реализует это желание:

gpgcheck=0

Конечно, если вы включите **gpgcheck**, для любой проверки **GPG** требуется ключ **GPG**; следующая директива указывает один ключ из локального каталога **/etc/pki/rpm-gpg** для этой цели:

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

Создайте свой собственный файл /etc/yum.repos.d

Вы хотите узнать, как создать локальный файл конфигурации в каталоге **/etc/yum.repos.d.** Он позволяет использовать команду **yum**, которая является самым простым способом установки групп пакетов, таких как **веб-сервер Apache**, или любой из групп пакетов, обсуждаемых в книге.

Для этого вам нужно настроить текстовый файл с расширением **.repo** в каталоге **/etc/yum.repos.d**. Все, что нужно файлу, это три строки. На самом деле, если вы готовы принять некоторые несмертельные ошибки, достаточно двух строк.

На RHEL 7, особенно во время экзамена, каталог **/etc/yum.repos.d** может быть пустым. Таким образом, у вас может не быть доступа к примерам, например, доступным для **CentOS**, как показано на **рисунке 7-7**. Первое руководство исходит из следующих комментариев в нижней части Файл **/etc/yum.conf**, который подтверждает, что файл должен иметь расширение **.repo** в **/etc/yum.repos.d** каталог:

 # PUT YOUR REPOS HERE OR IN separate files named file.repo
 # in /etc/yum.repos.d

Кроме того, вы можете настроить три строки в файле **/etc/yum.conf**. Если вы забыли, какие три строки добавить, на странице **man** для файла **yum.conf** приведен пример, показанный на **рис. 7-8.**

РИСУНОК 7-8 Выдержка из man yum.conf для настройки нового репозитория

```
[repository] OPTIONS
       The repository section(s) take the following form:

              Example: [repositoryid]
              name=Some name for this repository
              baseurl=url://path/to/repository/

              repositoryid  Must  be  a  unique  name for each repository, one
              word.

              name A human readable string describing the repository.

              baseurl Must be a URL to the directory  where  the  yum  reposi-
              tory's  `repodata' directory lives. Can be an http://, ftp:// or
              file:// URL. You can specify multiple URLs in one baseurl state-
              ment. The best way to do this is like this:
              [repositoryid]
              name=Some name for this repository
              baseurl=url://server1/path/to/repository/
                      url://server2/path/to/repository/
                      url://server3/path/to/repository/

              If you list more than one baseurl= statement in a repository you
              will find yum will ignore the  earlier  ones  and  probably  act
              bizarrely. Don't do this, you've been warned.
```

Если вы забыли, что делать, выполните команду **man yum.conf** и прокрутите вниз до этой части справочной страницы. Идентификатор хранилища указан в скобках. Если это не указано экзаменом **RHCSA**, не имеет значения, какое слово вы поместите в скобки в качестве идентификатора.

Для целей этой главы мы открываем новый файл с именем любое_имя.repo в директории **/etc/yum.repos.d**. (В некоторой степени имя файла **.repo** не имеет значения, если оно имеет расширение **.repo** в каталоге **/etc/yum.repos.d**.) В этом файле мы добавляем следующий идентификатор:

[test]

Далее идет директива имени для хранилища. Как видно из листинга на странице руководства, это имя должно быть «читаемым человеком». На языке Linux это также означает, что имя не влияет на функциональность репозитория. Для демонстрации добавим следующую директиву:

name=somebody likes Linux

!!! EXAM watch
Вы должны научиться создавать рабочий файл .repo в / etc / yum
Каталог .repos.d во время экзаменов Red Hat. Это может сэкономить много времени, когда вам нужно установить дополнительные пакеты.
!!!!!

Наконец, есть директива **baseurl**, которую можно настроить для указания на сервер установки. Требования **RHCSA** подразумевают, что вам нужно знать, как установить **Linux** с удаленного сервера. Они также предполагают, что вам нужно знать, как устанавливать и обновлять пакеты из удаленного хранилища. Для достижения любой цели

вам нужно знать **URL-адрес** этого удаленного сервера или хранилища. Разумно ожидать, что этот **URL** будет предоставлен во время экзамена. В главе 1 вы создали серверы **установки FTP и HTTP** в хост-системе для виртуальных машин, которые «удалены» от этих систем.

Установочные серверы **FTP и HTTP**, созданные вами в главе 1, также можно использовать в качестве удаленных репозиториев. Чтобы настроить доступ к этим репозиториям, все, что вам нужно включить, - это одна из следующих директив **baseurl**:

baseurl=ftp://192.168.122.1/pub/inst
baseurl=http://192.168.122.1/inst

Как предлагается на справочной странице **yum.conf**, вы не должны включать оба **URL-адреса в отдельные директивы baseurl**. Сделайте выбор и сохраните полученный файл. Это все, что вам нужно. Нет никаких оснований (кроме большей безопасности) включать директиву **enabled, gpgcheck или gpgkey**, описанную ранее. Конечно, безопасность важна в реальной жизни, но если вы сосредотачиваетесь на экзамене, лучший совет часто состоит в том, чтобы все было как можно проще.

После сохранения файла выполните следующие команды, чтобы сначала очистить базы данных из ранее использовавшихся репозиториев, а затем обновить кэш локальной базы данных из репозитория, вновь настроенного в файле **/etc/yum.repos.d/whwhat.repo:**

 # yum clean all
 # yum makecache

Для системы, не зарегистрированной в Red Hat Subscription Management, это приведет к следующему выводу:

Loaded plugins: langpacks, product-id, subscription-manager
test | 3.7 kB 00:00
test/primary_db | 2.9 MB 00:00
Metadata Cache Created

Теперь система готова к установке новых пакетов. Попробуйте выполнить следующую команду:

 # yum install system-config-date

Учитывая виртуальные машины, настроенные ранее в этой книге, вы можете увидеть результат, показанный на рисунке 7-9. В случае подтверждения команда **yum** загрузит, а затем установит не только RPM-файл **system-config-date**, но и зависимый пакет, показанный на рисунке, чтобы убедиться, что пакет **system-config-date** полностью поддерживается.

РИСУНОК 7-9 Установка одного пакета может включать зависимости.

```
Dependencies Resolved

================================================================================
 Package                    Arch          Version              Repository     Size
================================================================================
Installing:
 system-config-date         noarch        1.10.6-2.el7         test          619 k
Installing for dependencies:
 system-config-date-docs    noarch        1.0.11-4.el7         test          527 k

Transaction Summary
================================================================================
Install  1 Package (+1 Dependent package)

Total download size: 1.1 M
Installed size: 3.5 M
Is this ok [y/d/N]: 
```

УПРАЖНЕНИЕ 7-1
Создайте репозиторий yum из DVD RHEL 7

Для этого упражнения требуется доступ к DVD RHEL 7. Если у вас недостаточно места для этого упражнения, допустимо установить репозиторий непосредственно на смонтированном DVD. Кроме того, вы можете скопировать содержимое в указанный каталог. Он также предполагает наличие доступного репозитория установки, такого как один из созданных в Главе 1.

В этом упражнении предполагается, что вы начнете без файлов в каталоге **/etc/yum.repos.d**, описанном в этой главе.

1. Если в каталоге **/etc/yum.repos.d** есть существующие файлы, скопируйте их в папку резервной копии, например домашний каталог пользователя **root, /root.** Удалите все существующие файлы **.repo** в каталоге **/etc/yum.repos.d**.
   # cp -a /etc/yum.repos.d /root/
   # rm –f /etc/yum.repos.d/*.repo
2. Смонтируйте **DVD-диск RHEL 7** в каталог **/mnt** с помощью следующей команды (вам может понадобиться заменить **/dev/sr0** или **/dev/dvd на /dev/cdrom**):
   # mount /dev/cdrom /mnt
   В качестве альтернативы, если у вас есть только **RHEL 7 DVD как файл ISO**, подключите его с помощью следующей команды:
   # mount -o loop rhel-server-7.0-x86_64-dvd.iso /mnt
   Конечно, при желании вы можете скопировать файлы из другой точки монтирования, например, из **/mnt** в каталог **/opt/repos/rhel7**, с помощью следующей команды:
   # mkdir -p /opt/repos/rhel7
   # cp -a /mnt/. /opt/repos/rhel7
   Точка (.) Перед каталогом **/mnt** обеспечивает копирование содержимого каталога, а не самого каталога.
3. Перейдите в каталог **/etc/yum.repos.d.**
4. Откройте новый файл в текстовом редакторе. Используйте имя, такое как **rhel7.repo.**
5. Отредактируйте файл **rhel7.repo**. Создайте новый раздел директив. Используйте соответствующий заголовок строфы, например **[rhel].**
6. Укажите соответствующую директиву имени для хранилища.

7. Включите директиву **baseurl**, установленную в **file:///opt/repos/rhel7/**. Включите директиву **enabled=1**.
8. Сохраните и закройте файл.
9. Предполагая, что вы используете RHEL 7 (а не перестраиваемый дистрибутив), откройте файл **subscription-manager.conf** в каталоге **/etc/yum/pluginconf.d и** установите **enabled=0**.
10. Запустите команды **yum clean all и yum update.**
11. В случае успеха вы должны увидеть следующий результат:
    Loaded plugins: langpacks, product-id
    rhel | 3.8 kB 00:00:00
    (1/2): rhel/group_gz | 133 kB 00:00:00
    (2/2): rhel/primary_db | 3.4 MB 00:00:00
    No packages marked for update
    Теперь вы создали репозиторий в локальном каталоге **/opt/repos/rhel7**.
12. Восстановите исходные файлы. Откройте файл **subscription-manager.conf** в каталоге **/etc/yum/pluginconf.d** и затем установите **enabled=1**. Переместить резервные файлы из **/root /etc/yum.repos.d**. Если вы хотите восстановить исходную конфигурацию, удалите или переместите файл **rhel7.repo** из этого каталога. Запустите команду **yum clean all**.

Сторонние репозитории

Другие группы сторонних разработчиков создают пакеты для RHEL 7. Они включают пакеты для некоторых популярных программ, не поддерживаемых Red Hat. Веб-сайты для двух из этих третьих сторон можно найти по адресу **https://fedoraproject.org/wiki/EPEL и http://repoforge.org.**

Чтобы добавить сторонние репозитории в систему, вы должны создать собственный файл **.repo в /etc/yum.repos.d** каталог.

Некоторые репозитории, такие как **EPEL** (Дополнительные пакеты для Enterprise Linux(Extra Packages for Enterprise Linux)), упрощают настройку, предоставляя пакет RPM, который включает файл конфигурации .repo и ключ GPG для проверки пакетов. Чтобы настроить хранилище, все, что вам нужно сделать, это установить этот RPM-файл:

 # rpm -ivh https://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm

Если вы хотите отключить любой репозиторий в каталоге **/etc/yum.repos.d**, добавьте следующую директиву в соответствующий файл репозитория:

enabled=0

Основные команды yum

Если вы хотите узнать больше о тонкостях команды **yum**, запустите команду самостоятельно. Вы увидите следующую прокрутку вывода, вероятно, слишком быстро. Конечно, вы можете передать вывод на командный пейджер с помощью **yum | less** команд.

 # yum
Loaded plugins: langpacks, product-id, subscription-manager
You need to give some command
usage: yum [options] COMMAND
List of Commands
...

В следующих разделах вы узнаете, как работают некоторые из этих команд и параметров. Хотя у вас не будет доступа к Интернету во время экзамена Red Hat, у вас может быть сетевое подключение к локально настроенному хранилищу, которое вы должны быть готовы настроить через соответствующий файл в каталоге **/etc/yum.repos.d**, так как описано ранее. Как и во время экзамена, **yum** является отличным инструментом для администрирования систем **Red Hat**.

Начните с простой команды: **yum list**. Он вернет список всех пакетов, независимо от того, установлены они или доступны, а также номера версий и репозитории. **yum list | grep packagename** сообщает вам, какую версию пакета вы получите при установке **yum**. Если вы хотите показать все настроенные репозитории, вы можете сделать это с помощью **yum repolist all**. Дополнительную информацию о конкретном пакете можно получить с помощью команды **yum info**. Например, следующая команда функционально эквивалентна **rpm -qi samba**:

 # yum install samba

Команда **rpm -qi** работает, если запрашиваемый пакет уже установлен. Команда **yum info** не подпадает под это ограничение.

Если вы просто хотите поддерживать пакеты в системе в актуальном состоянии, выполните команду **yum update packagename**. Например, если у вас уже **установлен RPM Samba**, следующая команда гарантирует, что он обновлен до последней версии:

 # yum update samba

Если вы не установили **Samba**, эта команда не добавит его в ваши установленные пакеты. Таким образом, команда **yum update** аналогична команде **rpm -F**.

Конечно, команда **yum** не полна без опций, которые могут удалить пакет. Первый прост, потому что он удаляет пакет **Samba** вместе с любыми зависимостями:

 # yum remove samba

Команда **yum update** сама по себе является мощной; если вы хотите убедиться, что все установленные пакеты обновлены до последних стабильных версий, выполните следующую команду:

 # yum update

Команда **yum update** может занять некоторое время, так как она связывается с **Red Hat Portal** или другими репозиториями. Возможно, потребуется загрузить текущую базу данных пакетов со всеми зависимостями. Затем он находит все пакеты с доступными обновлениями и добавляет их в список пакетов для обновления. Он находит все зависимые пакеты, если они еще не включены в список обновлений.

Что если вы просто хотите получить список доступных обновлений? Команда **yum list updates** может помочь в этом. Это функционально эквивалентно команде **yum check-update**.

Но что, если вы не совсем уверены, что установить? Например, если вы хотите установить программу чтения документов **Evince** и считаете, что в операционную команду входит термин «**evince**», тогда может помочь команда **yum whatprovides «* evince *».**

Либо для поиска всех экземпляров файлов с расширением **.repo** выполните следующую команду:

 # yum whatprovides "*.repo"

В нем перечислены все экземпляры пакетов с файлами, заканчивающимися расширением **.repo**, с соответствующим пакетом **RPM**. Подстановочный знак является обязательным, поскольку для опции **whatprovides** требуется полный путь к файлу. Он принимает частичные имена файлов; например, команда **yum whatprovides «/etc/systemd/*»** возвращает RPM, связанный с файлами в
Каталог **/etc/systemd**. Как только нужный пакет известен, вы можете приступить к
Команда **yum install packagename.**

!!!!!
Во многих случаях проблемы с *yum* можно решить с помощью *yum clean all*. команда. Если имеются недавние обновления пакетов Red Hat (или сторонних репозиториев), эта команда очищает текущий кэш заголовков, позволяя синхронизировать заголовки с настроенными репозиториями, не дожидаясь шести часов по умолчанию, прежде чем кэш будет автоматически очищен.
!!!!

Безопасность и yum

Цифровые подписи GPG могут проверять целостность и подлинность обновлений yum. Это та же система, которая была описана ранее в этой главе для пакетов RPM. В качестве примера рассмотрим вывод при первой установке нового пакета по сети на RHEL 7:

 # yum install samba

После загрузки пакетов вы увидите нечто похожее на следующие сообщения:

Importing GPG key 0xFD431D51:
Userid : "Red Hat, Inc. (release key 2) <security@redhat.com>"
Fingerprint: 567e 347a d004 4ade 55ba 8a5f 199e 2f91 fd43 1d51
Package : redhat-release-server-7.0-1.el7.x86_64 (@anaconda/7.0)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y
Importing GPG key 0x2FA658E0:
Userid : "Red Hat, Inc. (auxiliary key) <security@redhat.com>"
Fingerprint: 43a6 e49c 4a38 f4be 9abf 2a53 4568 9c88 2fa6 58e0
Package : redhat-release-server-7.0-1.el7.x86_64 (@anaconda/7.0)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Is this ok [y/N]: y

Если вы одновременно загружаете пакеты из других репозиториев, дополнительные ключи **GPG** могут быть представлены на утверждение. Как указано в последней строке, **N** является ответом по умолчанию; вам действительно нужно ввести **y**, чтобы приступить к загрузке и установке - не только ключа **GPG**, но и соответствующего пакета.

Вы можете заметить, что используемый ключ **GPG** находится в том же каталоге ключей, что и команда **rpm**, описанная ранее в этой главе.

Обновления и исправления безопасности

Red Hat ведет открытый список ошибок, классифицированных по версии RHEL, по адресу **https://rhn.redhat.com/errata**. Если у вас есть подписка **RHEL**, затронутые пакеты обычно становятся доступны через портал **Red Hat**; для компьютеров, подключенных к **RHSM**, все, что вам нужно сделать, это периодически запускать команду yum update. Этот список полезен для тех третьих сторон, которые используют исходный код **RHEL**, таких как **CentOS, Scientific Linux и даже Oracle Linux**; как правило, перестройки **RHEL** дают аналогичные ошибки вскоре после **Red Hat**.

Пакетные группы и yum

Команда **yum** может сделать больше. Может устанавливать и удалять пакеты в группах. Это группы, определенные в файле *** -comps-Server.x86_64.xml**, описанном в главе 2. Одно место для этого файла находится на DVD-диске RHEL 7 в подкаталоге **/repodata**. В начале большинства этих разделов вы увидите **XML-директивы <id> и <name>,** в которых перечислены два идентификатора для каждой из этих групп.

Но это большая работа, чтобы найти группу пакетов. Команда **yum** делает это проще. С помощью следующей команды вы можете определить доступные группы пакетов из настроенных репозиториев:

 # yum group list

Обратите внимание, как группы делятся на установленные и доступные группы. Некоторые из перечисленных групп могут представлять особый интерес, например «Базовый веб-сервер»(“**Basic Web Server**,), который вы будете использовать в главе 14. Чтобы узнать больше об этой группе, выполните следующую команду. Вывод показан на **рисунке 7-10**.

 # yum group info "Basic Web Server"

В **yum** есть два типа групп: **обычные группы**, которые включают стандартные пакеты **RPM**, и **группы среды**, которые состоят из других групп. «Базовый веб-сервер» на рис. 7-10 обозначен как «группа среды» и фактически представляет собой набор обычных групп. Группы среды и обычные группы связаны с идентификатором среды и идентификатором группы соответственно, что показано командой **yum group info**. Эти идентификаторы альтернативные имена для групп, без пробелов или заглавных букв, и они часто используются в файлах конфигурации **Kickstart**.

РИСУНОК 7-10 Пакеты в группе Basic Web Server

```
[root@server1 ~]# yum group info "Basic Web Server"
Loaded plugins: langpacks, product-id

Environment Group: Basic Web Server
 Environment-Id: web-server-environment
 Description: Server for serving static and dynamic internet content.
 Mandatory Groups:
    base
    core
    web-server
 Optional Groups:
   +backup-client
   +directory-client
    guest-agents
   +hardware-monitoring
   +java-platform
   +large-systems
   +load-balancer
   +mariadb-client
   +network-file-system-client
   +performance
   +perl-web
   +php
   +postgresql-client
   +python-web
   +remote-system-management
   +web-servlet
[root@server1 ~]#
```

Чтобы перечислить все группы, вы можете набрать:

 # yum group list hidden

Давайте получим некоторую информацию об одной из обычных групп:

 # yum group info "Remote Desktop Clients"

Обратите внимание, что все пакеты перечислены как «Дополнительные пакеты». Другими словами, они обычно не устанавливаются вместе с группой пакетов. Таким образом, предположим, что вам нужно было выполнить следующую команду:

 # yum group install "Remote Desktop Clients"

Ничего не будет установлено. Желаемые пакеты из этой группы пакетов должны быть специально названы для установки с помощью команд, подобных следующим:

 # yum install tigervnc

Но дополнительные пакеты не единственная категория. Следующая команда выводит список всех пакетов в группе пакетов «Сервер печати». Вывод показан на **рисунке 7-11.**

 # yum group info "Print Server"

РИСУНОК 7-11 Пакеты в группе Сервер печати

```
[root@server1 ~]# yum group info "Print Server"
Loaded plugins: langpacks, product-id

Group: Print Server
 Group-Id: print-server
 Description: Allows the system to act as a print server.
 Mandatory Packages:
   +cups
   +ghostscript-cups
 Default Packages:
   +foomatic
   +foomatic-filters
   +gutenprint
   +gutenprint-cups
   +hpijs
   +paps
[root@server1 ~]#
```

Пакеты в группе «**Сервер печати**» подразделяются на две другие категории. Обязательные пакеты всегда устанавливаются вместе с группой пакетов. Пакеты по умолчанию обычно устанавливаются вместе с группой пакетов; однако, определенные пакеты из этой группы могут быть исключены с помощью ключа **-x**. Например, следующая команда устанавливает два обязательных пакета и шесть пакетов по умолчанию:

 # yum group install "Print Server"

Напротив, следующая команда исключает пакеты **paps** и пакеты **gutenprint-cups** из списка устанавливаемых:

 # yum group install "Print Server" -x paps -x gutenprint-cups

После выполнения этой команды снова покажите список пакетов в группе пакетов сервера печати:

 # yum group info "Print Server"

Вывод показан на **рисунке 7-12**. Сравните это с **рисунком 7-11**. Вы заметите, что некоторые пакеты имеют знак равенства **(=)** впереди. Это означает, что соответствующий пакет был установлен с помощью команды установки **yum group**. И наоборот, знак минус **(-)** указывает на то, что пакет был исключен из установки и не будет установлен, если мы обновим или установим группу. Точно так же знак плюс **(+)** указывает, что пакет не установлен, но он будет добавлен в систему, если мы установим или обновим группу. Если маркера нет, пакет был установлен, но не как часть команды установки **yum group**.

Параметры команды **yum** не являются полными, если только нет команды, которая может обратить процесс в обратном направлении. Как следует из названия, команда **group remove** удаляет все пакеты из указанной группы пакетов:

 # yum group remove "Print Server"

Исключения невозможны с командой удаления **yum group**. Если вы не хотите удалять все пакеты, перечисленные в выводе команды, лучше всего удалить целевые пакеты по отдельности.

РИСУНОК 7-12 Пакеты после установки группы «Сервер печати»

```
[root@server1 ~]# yum group info "Print Server"
Loaded plugins: langpacks, product-id

Group: Print Server
 Group-Id: print-server
 Description: Allows the system to act as a print server.
 Mandatory Packages:
   =cups
   =ghostscript-cups
 Default Packages:
   =foomatic
   =foomatic-filters
   =gutenprint
   -gutenprint-cups
   =hpijs
   -paps
[root@server1 ~]#
```

Больше команд yum

Доступен ряд дополнительных команд, связанных с **yum**. Два из них могут представлять особый интерес для тех, кто готовится к **экзаменам Red Hat: yum-config-manager и yumdownloader**, которые могут отображать все текущие настройки для каждого репозитория, а также загружать отдельные пакеты **RPM**. Еще одна связанная команда - **createrepo**, которая может помочь вам настроить локальный репозиторий.

Просмотреть все директивы с помощью yum-config-manager

В некоторой степени директивы, перечисленные в **yum.conf** и связанных файлах конфигурации, предоставляют лишь небольшой снимок доступных директив. Чтобы просмотреть полный список директив, выполните команду **yum-config-manager**. Перенаправте вывод команды утилите **less**. Сообщение включает в себя более 300 строк. Выдержка из раздела **[main],** показанного на **рис. 7-13**, содержит настройки, которые применяются ко всем настроенным репозиториям.

РИСУНОК 7-13 Частичный список директив yum

```
fssnap_automatic_keep = 1
fssnap_automatic_post = False
fssnap_automatic_pre = False
fssnap_devices = !*/swap,
   !*/lv_swap
fssnap_percentage = 100
gaftonmode = False
gpgcheck = True
group_command = objects
group_package_types = mandatory,
   default
groupremove_leaf_only = False
history_list_view = single-user-commands
history_record = True
history_record_packages = yum,
   rpm
http_caching = all
installonly_limit = 3
installonlypkgs = kernel,
   kernel-bigmem,
   installonlypkg(kernel-module),
   installonlypkg(vm),
   kernel-enterprise,
   kernel-smp,
   kernel-debug,
   kernel-unsupported,
   kernel-source,
   kernel-devel,
   kernel-PAE,
   kernel-PAE-debug
```

Некоторые из директив, связанных с **yum**, не заполнены, такие как **exactarchilist**; некоторые не имеют значения, например, директивы о цвете. Некоторые из других важных директив показаны в **Таблице 7-5**. Это не полный список. Если вас интересует не показанная директива, она определена на странице руководства для файла **yum.conf**.

yum-config-manager также может управлять репозиториями. Например, если вам известен URL-адрес хранилища, вы можете автоматически создать файл конфигурации с помощью команды, подобной следующей:

 # yum-config-manager --add-repo=http://192.168.122.1/inst

Загрузка пакетов с помощью yumdownloader

Как следует из названия, команда **yumdownloader** может использоваться для загрузки пакетов из репозиториев на основе **yum**. Это довольно простая команда. Например, следующая команда просматривает содержимое настроенных репозиториев для пакета с именем cups:

 # yumdownloader cups

ТАБЛИЦА 7-5 Параметры конфигурации из yum-config-manager

Директива конфигурации в yum	Описание
alwaysprompt	Запрашивает подтверждение установки или удаления пакета.
assumeyes	По умолчанию установлено значение no; если установлено значение 1, **yum** автоматически выполняет установку и удаление пакета.
cachedir	Установите каталог для базы данных и загруженных файлов пакета.

distroverpkg	Перечисляет пакеты **RPM**, которые yum проверяет, чтобы найти версию дистрибутива Linux, установленную на текущем компьютере.
enablegroups	Поддерживает команды **yum group *.**
installonlypkgs	Перечисляет пакеты, которые никогда не должны обновляться; обычно включает в себя пакеты ядра Linux.
logfile	Определяет имя файла с информацией журнала, обычно **/var/log/yum.log**.
pluginconfpath	Отмечает каталог с плагинами, обычно **/etc/yum/pluginconf.d**.
reposdir	Определяет каталог с файлами конфигурации репозитория.
ssl*	Поддерживает использование **Secure Sockets Layer (SSL)** для безопасных обновлений.
tolerant	Определяет, останавливается ли **yum** в случае возникновения ошибки с одним из пакетов

Либо пакет **RPM** загружается в локальный каталог, либо команда возвращает следующие сообщения об ошибках:

No Match for argument cups
Nothing to download

Иногда требуется больше подробностей. Если в хранилище хранится несколько версий пакета, по умолчанию загружается последняя версия этого пакета. Это не всегда может быть тем, что вы хотите. Например, если вы хотите использовать изначально выпущенное ядро RHEL 7, используйте следующую команду:

 # yumdownloader kernel-3.10.0-123.el7

Создайте свой собственный репозиторий с createrepo

В более ранней версии задачей RHCE для RHEL 6 предлагалось, чтобы вы знали, как «**создать частный репозиторий yum**». Хотя эта задача с тех пор была удалена, это необходимый рабочий навык для системного инженера **Red Hat**.

Пользовательские репозитории могут обеспечить дополнительный контроль. Предприятия, которые хотят управлять пакетами, установленными в их системах **Linux**, могут создавать свои собственные настраиваемые хранилища. Хотя это может быть основано на стандартных репозиториях, разработанных для дистрибутива, оно может включать в себя дополнительные пакеты, такие как пользовательское программное обеспечение, уникальное для организации. Также легко можно пропустить пакеты, которые могут нарушать организационные политики, такие как игры. Ограничения выбора для определенных функций, таких как браузеры, могут минимизировать связанные требования поддержки.

ЦЕЛЬ СЕРТИФИКАЦИИ 7.04

олнительные инструменты управления пакетами

Независимо от того, подключена ли система **Red Hat** к клиентскому порталу **Red Hat** или к репозиториям, предоставляемым дистрибутивом, таким как **CentOS** или

Scientific Linux, она будет использовать одни и те же базовые инструменты управления пакетами. Независимо от источника, команда **rpm** используется для обработки пакетов **RPM**. Инструменты более высокого уровня, такие как команда **yum**, используются для удовлетворения зависимостей и установки групп пакетов. Это имеет смысл, поскольку дистрибутивы перестроения основаны на том же исходном коде, что и **RHEL 7**, и все они распространяются через **RPM**.

Эти сходства распространяются на инструменты управления пакетами на основе графического интерфейса. Хотя идентичность этих инструментов изменилась между **RHEL 6 и RHEL 7**, они по-прежнему являются интерфейсом команд **rpm** и **yum**. Они используют преимущества групп пакетов, настроенных в **XML-файле**, описанном в **главе 2**. Поскольку **Red Hat** использует **GNOME** в качестве среды рабочего стола с графическим интерфейсом по умолчанию, соответствующие инструменты управления программным обеспечением основаны на этом интерфейсе.

В этой среде соответствующие инструменты управления программным обеспечением основаны на этом интерфейсе. В **RHEL 7** инструменты управления пакетами на основе графического интерфейса опираются на **PackageKit**, общий уровень абстракции, который обеспечивает унифицированный интерфейс для всех приложений управления программным обеспечением **Linux**. Однако вполне возможно, что **PackageKit** не будет доступен на сервере или даже в системе, настроенной на экзамен **Red Hat**. Если вам абсолютно необходим **PackageKit**, установите необходимые **RPM** с помощью команды yum **install gnome-packagekit**. Конечно, если вы уже знакомы с командой **yum**, вам может не понадобиться **PackageKit**.

!!!!
Хотя *RHN* указан как часть целей *RHCSA*, в контексте он указан как выбор. Независимо от того, устанавливаете ли вы или обновляете пакеты программного обеспечения из *RHN*, «*удаленного хранилища или локальной файловой системы*», вы можете использовать команды *rpm* и *yum*. Конечно, проще всего, если у вас есть официальная подписка на Red Hat.
!!!!

Средство обновления программного обеспечения GNOME

Средство обновления программного обеспечения можно запустить из терминала с графическим интерфейсом с помощью команды **gpk-update-viewer**. Либо в среде рабочего стола **GNOME** щелкните **Приложения | Системные инструменты | Обновление программного обеспечения**. Инструмент, как показано на **рисунке 7-14**, перечисляет пакеты, которые доступны для обновления.

Это довольно простой интерфейс. Это эффективный интерфейс для команды обновления **yum**. Обратите внимание на дополнительную информацию, с описанием изменений.

Автоматические обновления

Важно установить последние обновления безопасности как можно быстрее. Для этого откройте инструмент «**Настройки обновления программного обеспечения**», показанный **на рис. 7-15**. Вы можете открыть его из командной строки графического интерфейса с помощью команды **gpk-prefs**. Вы можете настроить систему на проверку обновлений **ежечасно, ежедневно или еженедельно**, или не делать этого вообще. Когда обновления найдены, вы можете настроить автоматическую установку всех доступных обновлений, только обновлений безопасности или вообще ничего.

РИСУНОК 7-14 Инструмент обновления программного обеспечения GNOME

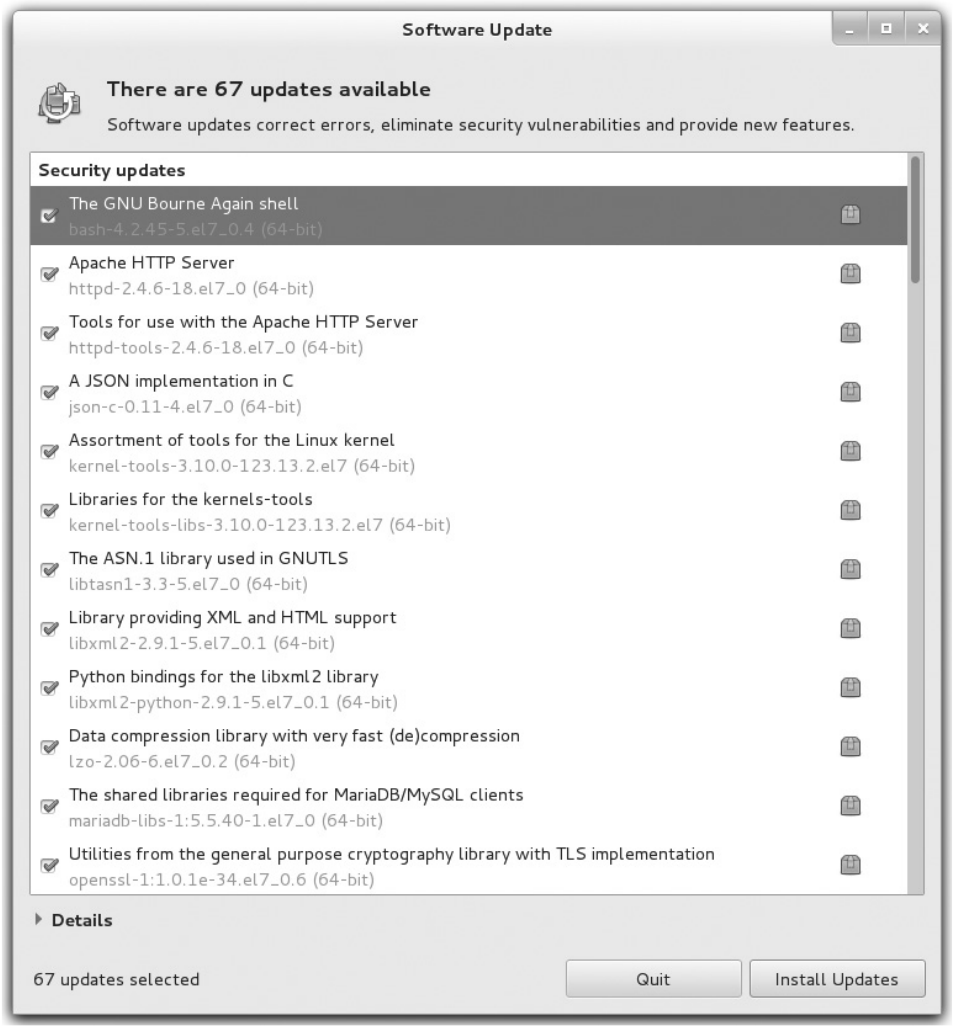
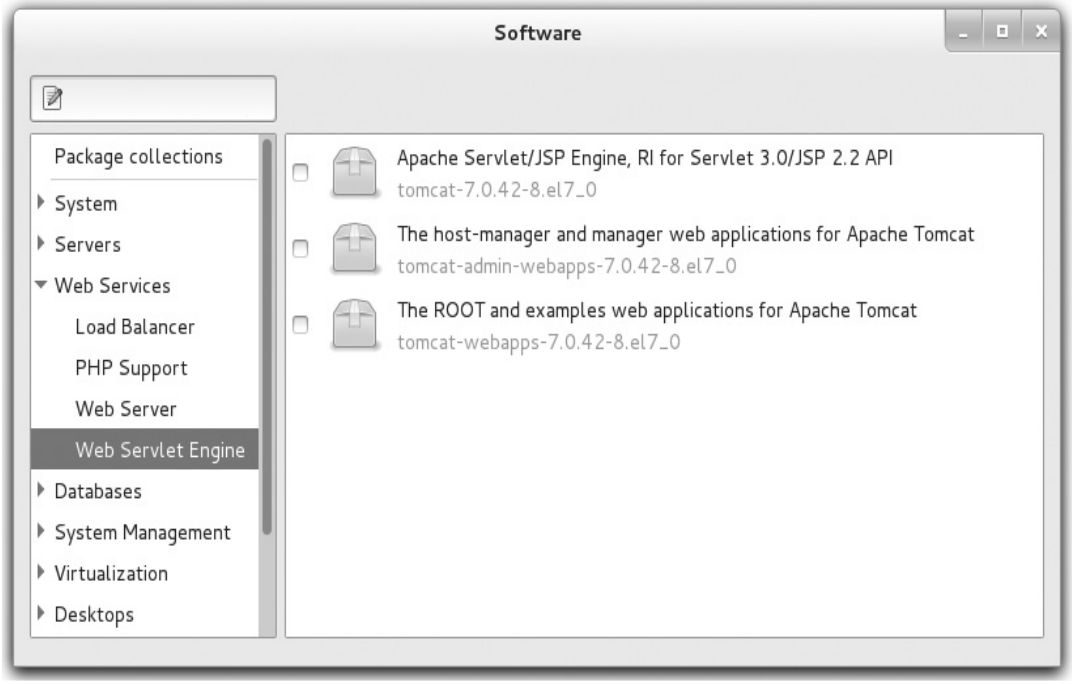


РИСУНОК 7-15. Инструмент настроек обновления программного обеспечения GNOME

GNOME Software Tool

Вы можете добавлять, обновлять и удалять пакеты с помощью графического инструмента. Чтобы запустить инструмент **GNOME Software** из командной строки графического интерфейса пользователя, выполните команду **gpk-application** или щелкните **Приложения | Системные инструменты | Программного обеспечения**. Откроется инструмент, показанный на **рисунке 7-16**. Здесь вы можете установить более одного пакета или группы пакетов одновременно. После того, как пакеты выбраны (или отмена выбора), инструмент автоматически вычисляет зависимости и устанавливает (или удаляет) их вместе с выбранными пакетами.

РИСУНОК 7-16 Программный инструмент GNOME



Вы можете использовать программный инструмент **GNOME** для добавления **пакетов или групп пакетов** по вашему выбору. В верхней левой части экрана находится опция «**Коллекции пакетов**», в которой перечислены те же группы, что и в выводе команды **yum group list**, описанной ранее.

Пакеты программ дополнительно подразделяются в нижней левой части экрана. Когда пакет или группа пакетов выбраны или отменены для установки или удаления, кнопка «**Применить изменения**» становится активной. После нажатия инструмент использует команду **yum** для вычисления зависимостей. Если нет никаких зависимостей, установка продолжается немедленно. Если есть зависимости, полный список пакетов, которые будут установлены или удалены, представлен на ваше утверждение.

УПРАЖНЕНИЕ 7-2
Установка More с yum и программным инструментом GNOME

Для этого упражнения требуется сетевое подключение к удаленному репозиторию или, по крайней мере, DVD-диск RHEL 7, скопированный или смонтированный в качестве репозитория, как было настроено ранее в этой главе. Если вы используете пересборку RHEL 7, вам нужно убедиться, что соединение с основным репозиторием активно, возможно, с помощью команды **ping** для хоста этого репозитория, как определено в соответствующем файле в каталоге **/etc/yum.repos.d**. Учитывая возможные вариации, точные шаги невозможны.

1. Запустите команду **yum list**. Предполагая активное сетевое соединение и отзывчивый репозиторий, вы увидите полный список доступных пакетов, включая те, которые уже установлены. Обратите внимание на метку в правом столбце; он либо покажет хранилище, где пакет доступен, либо заметит, что пакет уже установлен.
2. Введите команду **gpk-application** в командной строке **GUI**. Это должно открыть инструмент программного обеспечения **GNOME**.
3. Во второй консоли командной строки введите команду **yum group list**. В инструменте программного обеспечения GNOME выберите Коллекции пакетов. Сравните список групп пакетов в каждом выводе.
4. Просмотрите доступные группы пакетов в программном инструменте **GNOME**. Например, нажмите стрелку рядом с Серверы. Под параметрами, которые появляются, нажмите **FTP-сервер**. В конфигурации RHEL 7 этой группы есть только два официальных пакета. Выберите пакеты, которые будут установлены при нажатии кнопки «**Применить изменения**».
5. Найдите текстовое поле в верхнем левом углу программного средства **GNOME**. Введите общий поисковый термин, такой как **gnome**, и просмотрите, как отображается длинный список пакетов. Сравните результат с выводом команды **yum search gnome**.
6. Используйте менее распространенный поисковый термин, такой как **iptables**. Выделите пакет **iptables** и просмотрите его в правой нижней части экрана. Сравните результат с выводом команды **yum info iptables**.
7. Снова выберите пакеты **iptables** и нажмите кнопки «**Файлы и зависимые пакеты**» в правой нижней части экрана. Сравните результаты с выводом команд **repoquery -l iptables и yum deplist iptables.**
8. Выбрав несколько пакетов, нажмите «**Применить изменения**». Если есть зависимости, они будут установлены автоматически.
9. Подождите, пока пакеты установлены. Когда закончите, закройте инструмент **GNOME Software.**

Диспетчер подписок Red Hat

Цели экзамена **RHCSA** требуют, чтобы кандидаты могли «устанавливать и обновлять пакеты программного обеспечения из **Red Hat Network**». Однако на момент написания статьи системы **RHEL 7** больше не поддерживали использование **Red Hat Network (RHN).** Подписки на репозитории **RHN** доступны только при локальной установке более старой версии **Red Hat Satellite**. Более новые версии **Red Hat Satellite Server и автономных систем RHEL 7** используют Red Управление подписками на порталы для клиентов Hat (**RHSM**) для доступа к репозиториям программного обеспечения **Red Hat**.

Помните, что связанная с этим задача предполагает, что все, что вам нужно знать, это как устанавливать и обновлять пакеты из **RHN**. И этот навык уже охватывался командами **rpm** и **yum**, а также соответствующими инструментами **GUI**, которые обсуждались в большей части этой главы.

Возможно, ключевым преимуществом **Red Hat Satellite** (или такой альтернативы, как **Spacewalk** или **Katello**) является возможность удаленного управления всеми **RHEL** и перестройка систем распределения через веб-интерфейс. После настройки соответствующего соединения из клиентских систем **Satellite Server** может даже запускать удаленные команды по любому расписанию. Если вы управляете целой группой систем, **Red Hat Satellite** поддерживает настройку систем в группах. Например, если есть 10 систем, настроенных как веб-серверы на основе **RHEL 7**, вы можете настроить эти системы как одну группу. Затем вы можете запланировать одну команду, которая будет

применена ко всем этим системам удаленно. Для получения дополнительной информации о **Red Hat Satellite** см. Последнюю версию документации, доступную по адресу **https://access.redhat.com/documentation/.**

Если у вас есть доступ к управлению подписками **Red Hat Customer Portal (RHSM),** выполните следующие действия, чтобы зарегистрироваться и подписаться на систему RHEL 7:

1. Выполните следующую команду, чтобы зарегистрировать систему в **RHSM**. Включите имя пользователя и пароль действительной учетной записи Red Hat. Если система уже зарегистрирована, используйте параметр команды **--force** для повторной регистрации машины.
   # subscription-manager --username=USERNAME --password=PASSWORD
   (# subscription-manager --username = ИМЯ ПОЛЬЗОВАТЕЛЯ --password = ПАРОЛЬ)
2. Подпишите систему на продукт **Red Hat**. В следующей команде параметр **--auto** находит наиболее подходящую подписку:
   # subscription-manager attach –auto
3. Кроме того, список всех доступных подписок. Обратите внимание на идентификаторы пула pool IDs.
   # subscription-manager list –available
   Затем используйте идентификатор пула, полученный из последней команды, чтобы подключить систему к подписке:
   # subscription-manager attach --pool=8a85f98146f719180146fd9593b7734c
4. Просмотрите текущие настройки. Выполните следующую команду, чтобы получить список подписок, подключенных к системе:
   # subscription-manager list
5. Показать все доступные репозитории для системы:
   # subscription-manager repos
6. Вы можете включить дополнительные репозитории с помощью следующей команды:
   # subscription-manager repos --enable=REPOID
7. Если вы предпочитаете, вы можете использовать **GUI-версию Subscription Manager**, которую можно запустить с помощью команды **subscription-manager-gui**. Либо в среде рабочего стола GNOME щелкните **Приложения | Системные инструменты | Диспетчер подписок Red Hat.**

РЕЗЮМЕ СЕРТИФИКАЦИИ

Эта глава посвящена управлению пакетами **RPM**. Используя различные параметры команд, вы также увидели, как средство **rpm** устанавливает, удаляет и обновляет пакеты, а также как оно работает локально и удаленно. Когда представлена новая версия ядра, важно никогда не заменять существующее ядро на **rpm**. Правильно настроенная установка более поздней версии ядра не перезаписывает, а помещает ядра рядом друг с другом. После этого вы сможете загрузиться в любое из ядер.

С помощью команды **rpm** вы также узнали, как запрашивать пакеты, проверять, к какому пакету принадлежит файл, проверять подпись пакета и находить текущий список установленных **RPM**. Вы также увидели трудности, связанные с зависимостями, которые привели пользователей к команде **yum**.

Отчасти команда **yum** является внешним интерфейсом команды **rpm**. Когда есть зависимости, он устанавливает эти пакеты одновременно. Вы узнали, как настроить **Red Hat** и другие репозитории для работы с командой **yum**. Теперь вы сможете настроить даже **DVD RHEL 7** в качестве собственного репозитория. Как вы видели, команда **yum** также может устанавливать или удалять группы пакетов, как определено в файле базы

данных XML пакетов на DVD-диске RHEL 7 и других репозиториях. Команда **yum** полностью совместима с **RHSM**.

Хотя в графическом интерфейсе доступны дополнительные инструменты управления пакетами, они являются интерфейсом команд **yum и rpm**. С помощью команды **gpk-update-viewer** вы запустили инструмент обновления программного обеспечения, чтобы определить и установить доступные обновления. С помощью команды **gpk-prefs** вы запустили инструмент «**Настройки обновления программного обеспечения**», который может регулярно проверять и устанавливать безопасность или все доступные обновления. С помощью команды **gpk-application** вы открыли инструмент **GNOME Software**, который также можно использовать для добавления или удаления пакетов и групп пакетов. Если у вас есть подписка **RHEL**, системы также могут обновляться и регистрироваться в **RHSM** с помощью команды **subscription-manager**.

ПАРУ МИНУТ ПРОВЕРКИ

Вот некоторые из ключевых моментов целей сертификации в главе 7.

Менеджер пакетов Red Hat(rpm)

- База данных RPM отслеживает, где находится каждый файл в пакете, его номер версии, и многое другое.
- Команда **rpm -i** устанавливает пакеты **RPM**.
- Команда **rpm -e** удаляет пакеты **RPM**.
- Команда **rpm** может даже устанавливать **RPM** непосредственно с удаленных серверов.
- Проверка пакета **RPM** поддерживается ключами **GPG** в **/etc/pki/rpm-gpg** каталог.
- **RPM** ядра всегда должны быть установлены, а не обновлены.
- Режим обновления **RPM** заменяет старую версию пакета новой.

Больше RPM команд

- Команда **rpm -q** определяет, установлены ли пакеты в системе; с дополнительные переключатели, он может перечислить больше о пакете и определить пакет для конкретного файла.
- Сигнатуры пакетов можно проверить с помощью команды **rpm --checksig (или -K)**.
- Команда **rpm -V** может идентифицировать файлы, которые изменились с оригинала установка пакета до установки **RPM**.
- Команда **rpm -qa** выводит список всех установленных на данный момент пакетов.

Зависимости и команда yum

- Включая дополнительные обязательные пакеты, команда **yum** может помочь избежать «Ад зависимость».
- Поведение команды **yum** настраивается в файле **/etc/yum.conf**, плагины в каталоге **/etc/yum/pluginconf.d** и репозиториях, настроенных в **/etc/yum.repos.d** каталоге.
- **Red Hat** организует пакеты в нескольких разных репозиториях для **RHEL 7**.
- Репозитории для восстановления дистрибутивов и от третьих лиц доступны онлайн.
- Команда **yum** может устанавливать, удалять и обновлять пакеты. Это также может быть использовано для поиска других путей.
- Команда **yum** использует ключи **GPG**, разработанные для пакетов **RPM**.

- Команда **yum** может устанавливать, удалять и перечислять группы пакетов.

Дополнительные инструменты управления пакетами

- Инструменты управления пакетами **RHEL 7** основаны на **PackageKit**, созданном для **GNOME**.
- С помощью инструмента **GNOME Software** вы можете устанавливать и удалять пакеты и пакеты групп.
- **PackageKit** также включает в себя инструменты, ориентированные на текущие обновления. Также можно настроить обновления пакетов безопасности или всех пакетов по расписанию.
- **RHSM и Red Hat Satellite** могут помочь вам удаленно управлять подписанными системами используя веб-интерфейс.

Самопроверка

Следующие вопросы помогут оценить ваше понимание материалов, представленных в этой главе. Поскольку на экзаменах **Red Hat** не появляется вопросов с несколькими вариантами ответов, вопросы с несколькими вариантами ответов не отображаются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Это нормально, если у вас есть другой способ выполнения задачи. Получение результатов, а не запоминание пустяков - вот на что рассчитывает **Red Hat** Экзамены. На многие из этих вопросов может быть более одного ответа.

Менеджер пакетов Red Hat

1. Какую команду вы бы использовали для установки пакета penguin-3.26.x86_64.rpm, с дополнительным сообщения в случае ошибок? Пакет находится в локальном каталоге.

_____

2. Какой командой вы воспользуетесь, чтобы обновить RPM-версию penguin с помощью **penguin-3.27.x86_64.rpm** пакет? Пакет находится на сервере ftp.remotemj02.abc.

_____

3. Если вы загрузили более позднюю версию ядра Linux в локальный каталог и пакет имя файла **kernel-3.10.0-123.13.2.el7.x86_64.rpm**, какая команда лучше всего сделать его частью ваша система?

_____

4. Какой каталог содержит ключи RPM GPG в установленной системе?

_____

Больше RPM команд

5. Какая команда перечисляет все установленные в настоящее время RPM?

_____

6. Какая команда перечисляет все файлы в пакете penguin-3.26.x86_64.rpm?

_____

7. Если вы загрузили RPM от третьей стороны с именем third.i686.rpm, как вы можете проверить подпись связанного пакета?

_____

Зависимости и команда yum

8. Какой полный путь к каталогу, где обычно находятся сконфигурировано репозитории yum?

_____

9. Какая команда ищет в репозиториях **yum** пакет, связанный с файлом /etc/passwd?

_____

Дополнительные инструменты управления пакетами

10. Какая команда командной строки выводит список групп пакетов, показанных в программном инструменте GNOME?

_____

11. Назовите два допустимых периода времени для автоматического обновления, как определено в Обновлении программного обеспечения. **Инструмент настроек.**

_____

_____

12. Какая команда из консоли запускает процесс регистрации на RHSM?

_____

LAB ВОПРОСЫ

Red Hat представляет свои экзамены в электронном виде. По этой причине лабораторные работы в этой главе доступны с DVD, который сопровождает книгу в подкаталоге Chapter7 /. Они доступны в .doc, .html, и формат .txt для отражения стандартных параметров, связанных с электронной доставкой в действующей системе RHEL 7.

Если вы еще не настроили RHEL 7 в системе, обратитесь к первой лабораторной главе главы 2 для установки. инструкции. Ответы для каждой лаборатории следуют за ответами самопроверки для вопросов, которые заполняются.

Labs

Во время экзаменов Red Hat задания будут представлены в электронном виде. Таким образом, эта книга также представляет большинство лабораторий в электронном виде. Для получения дополнительной информации см. Раздел «Лабораторные вопросы» в конце главы 7.

Лаборатория 1

В этой лабораторной работе предполагается, что вы завершили лабораторную работу 2 в **главе 1**. Хотя лучше иметь работающее сетевое подключение к этой системе, безусловно, можно настроить его на той же системе, что и в лабораторной работе, при условии, что соответствующий сервер работает В любом случае вам может потребоваться

убедиться, что какой-либо брандмауэр не работает (или, что лучше, правильно настроен в соответствии с главой 4). Подсказка: в соответствии с главой 1, лабораторная работа 2, вы будете подключаться к хранилищу на основе **FTP-сервера по IP-адресу 192.168.122.1**.

Вы настроите **yum** для установки пакетов из репозитория, который вы сделали доступным в **Главе 1**, на вашем главном сервере. Создайте файл **file.repo** в каталоге **/etc/yum.repos.d**. Можно использовать любые существующие файлы в этом каталоге в качестве модели. Используйте **[inst]** в качестве идентификатора и установите имя репозитория. Для справки: файлы установки, созданные в этой лаборатории, были скопированы в подкаталог **pub/inst/**. Убедитесь, что локальная система действительно ищет хранилище и использует правильный файл для проверки.

Лаборатория 2

В этой лабораторной работе вы будете определять потенциальную проблему безопасности. Сотрудники службы безопасности сообщили вам, что проблема связана с двоичным файлом, который запускает сервер.

В этой лабораторной работе вы будете использовать скрипт в каталоге Chapter7/DVD-диска, входящего в комплект книги. Скрипт называется **Ch7Lab2**. Вам необходимо скопировать сценарий в тестовую систему **server1.example.com**, созданную в главе 2. В следующих шагах предполагается, что система находится на виртуальной машине на основе KVM.

1. Откройте диспетчер виртуальных машин из командной строки графического интерфейса с помощью команды **virt-manager**.
2. Подключитесь к локальной системе **(QEMU).**
3. Дважды щелкните виртуальную машину с системой server1.example.com. В появившемся окне нажмите **View | Подробности**.
4. Вставьте DVD для книги. Используйте отображаемые параметры для подключения дисковода CD / DVD к виртуальной машине. Вернитесь в консоль для виртуальной машины, нажав View | consol.
5. Загрузите систему **server1.example.com**. Смонтируйте DVD книги с помощью команды **mount /dev/cdrom/media.**
6. Войдите в систему с учетной записью администратора.
7. Убедитесь, что вы находитесь в каталоге **/root** с помощью команды **cd /root.** Скопируйте указанный сценарий (и связанный тестовый файл) с DVD-диска в домашний каталог пользователя root с помощью команды **cp /media/Chapter7/Ch7Lab2 ~**.
8. Убедитесь, что скрипт исполняемый; выполните команду **chmod u + x Ch7Lab2**.
9. Как пользователь root выполните сценарий с помощью команды ./**Ch7Lab2**.

Теперь вы можете начать реальную работу лаборатории.

Лаборатория 3

Эта лабораторная работа может оказаться невозможной, если обновления не доступны из RHSM (или если вы используете пересборку RHEL 7, удаленного репозитория с обновлениями, настроенного в файлах в каталоге **/etc/yum.repos.d**). В этой лабораторной работе вы изучите, что происходит, когда вы запускаете обновление для обновления до новых версий пакетов, доступных для новых функций, для решения проблем безопасности и многого другого. Перед началом выполните следующую команду, чтобы очистить кеш, чтобы включить полный набор сообщений:

 # yum clean all

Запустите следующую команду и просмотрите вывод:

 # yum update

Если доступно много обновлений, этот процесс может занять некоторое время. Если вы хотите загрузить и установить обновления, используйте ключ **-y**, который отвечает «да» на все запросы. Сохраните вывод в файл. Полная команда становится

 # yum update -y> update.txt

После завершения загрузки и установки просмотрите файл update.txt. Обратите внимание на первые сообщения о том, как плагины загружаются из каталога **/etc/yum/pluginconf.d**. Обратите внимание, как он загружает информацию из репозиториев, загружает заголовки и разрешает зависимости.

Как только зависимости разрешены, проверьте, откуда происходят загрузки. Обратите внимание, как устанавливаются некоторые пакеты и как обновляются другие.

Лаборатория 4

В этой лабораторной работе вы создадите задание, которое автоматически проверяет наличие обновлений безопасности в локальной системе ежечасно. Хотя такую задачу можно настроить с помощью демона **cron**, это тема главы 9.

Лаборатория 5

В этой лабораторной работе вы изучите, что происходит, когда вы обновляете RPM ядра, устанавливая его рядом с существующим ядром. Если более новое ядро недоступно, используйте старое ядро, доступное на DVD-диске книги, в подкаталоге Chapter 7 /. Для этого в этом каталоге доступны два ядра, оба для 64-битных систем:

kernel-3.10.0-123.el7.x86_64.rpm
kernel-3.10.0-229.el7.x86_64.rpm

Соответствующий пакет прошивки linux также включен в каталог Chapter7 /, так как является зависимостью ядра. Просто помните, что если вам не нужно ядро, которое вы устанавливаете во время этой лабораторной работы, убедитесь, что правильно удалили пакет после завершения лабораторной работы.

1. Сделайте копию существующего файла конфигурации GRUB 2, **/boot/grub2/grub.conf**. Распечатайте его или скопируйте в свой домашний каталог.
2. Сделайте копию текущего списка файлов в каталоге **/boot**. Например, запустите команду **ls /boot> bootlist**, которая записывает список файлов в файл **bootlist**.
3. Если доступно более новое ядро, и вы подключены к RHSM или другому подходящему репозиторию, выполните следующую команду:

 # yum install kernel

Другой альтернативой является загрузка и установка RPM с помощью команды **yumdownloader**.

4. Проверьте результаты в файле конфигурации GRUB 2, **/boot/grub2/grub.conf**. Обратите внимание на отличия от старого файла конфигурации GRUB 2, который вы сохранили на

шаге 1. Что такое ядро по умолчанию? Если вам довелось установить старое ядро, по умолчанию вы ожидаете этого?
5. Проверьте результаты в каталоге **/boot.** Обратите внимание на различия с исходным списком файлов в каталоге **/boot**. Проверьте результат с перезагрузкой.

Лаборатория 6

В этой лабораторной работе вы будете использовать как команду **yum**, так и программный инструмент GNOME для установки пакетов из группы пакетов клиентов удаленного рабочего стола. На данный момент, несколько виртуальных машин должны быть доступны. Обязательно установите все пакеты из группы пакетов клиентов удаленного рабочего стола в системах **server1.example.com и tester1.example.com**. В одной системе используйте инструмент программного обеспечения **GNOME**. Во второй системе используйте команду **yum**.

Группа пакетов «Клиенты удаленного рабочего стола» находится в категории «Рабочий стол» в программном обеспечении GNOME. Вы также можете использовать определенную команду **yum** для этой цели. Убедитесь, что каждый пакет, указанный в группе, установлен в обеих системах.

ТЕСТОВЫЕ ОТВЕТЫ

Менеджер пакетов Red Hat

1. Команда, устанавливающая пакет penguin-3.26.x86_64.rpm, с дополнительными сообщениями в случае ошибки, из локального каталога есть
 # rpm -iv penguin-3.26.x86_64.rpm
Дополнительные ключи, которые не изменяют функциональность команды, например -h для хэша оценки, являются приемлемыми. Это относится и к последующим вопросам.

2. Команда, которая обновляет вышеупомянутую RPM-версию пингвина с помощью penguin-3.27.x86_64 Пакет .rpm с сервера ftp.remotemj02.abc

 # rpm -Uv ftp://ftp.remotemj02.abc/penguin-3.26.x86_64.rpm

Если вы используете сервер vsFTP по умолчанию, пакет может находиться в подкаталоге pub / Packages /. Другими словами, команда будет
 # rpm -Uv ftp://ftp.remotemj02.abc/pub/Packages/penguin-3.26.i386.rpm

Да, вопрос не точный, но это то, что вы видите в реальной жизни.

3. Если вы загрузили более позднюю версию ядра Linux в локальный каталог и пакет имя файла - **kernel-3.10.0-123.13.2.el7.x86_64.rpm**, лучший способ сделать его частью вашей системы это установить его, а не обновлять текущее ядро. Обновления ядра перезаписывают существующие ядра.
Установки ядра позволяют ядрам существовать бок о бок; если новое ядро не работает, вы все равно можете загрузиться в рабочее ядро. Поскольку нужный пакет уже загружен, вы используете команду похож на следующее:

 # rpm -iv kernel-3.10.0-123.13.2.el7.x86_64.rpm

Варианты команды rpm, такие как rpm -i и rpm -ivh, являются приемлемыми. Тем не менее, вариации это обновление с ключами -U или -F неверно.

4. Каталог с RPM-ключами RPM в установленной системе: **/etc/pki/rpm-gpg**. Ключи GPG на CD / DVD RHEL 7 не «установлены» в системе.

Больше RPM команд

5. Команда, в которой перечислены все установленные RPM,

 # rpm -qa

6. Команда, в которой перечислены все файлы в пакете **penguin-3.26.x86_64.rpm**,

 # rpm -ql penguin-3.26.x86_64.rpm

7. Если вы загрузили RPM от третьего лица, назовите его Third.i686.rpm, сначала вам нужно загрузите и установите файл RPM-GPG-KEY, связанный с этим хранилищем. Вы можете тогда проверьте подпись соответствующего пакета с помощью команды, подобной следующей (обратите внимание на верхний регистр -K; --checksig эквивалентно -K):

 # rpm -K third.i386.rpm

Зависимости и команда yum

8. Репозитории команд yum обычно настраиваются в файлах в каталоге **/etc/yum.repos.d**. Технически, репозитории команд yum также можно настроить непосредственно в файле **/etc/yum.conf**.

9. Команда **yum whatprovides /etc/passwd** или **yum provides /etc/passwd** определяет пакеты, связанные с этим файлом.

Дополнительные инструменты управления пакетами

10. Это немного сложный вопрос, потому что команда **yum group list** выводит список групп пакетов также отображается в программном инструменте GNOME.

11. Допустимые периоды времени для обновлений, определенные инструментом «Настройки обновлений программного обеспечения», **ежечасно, ежедневно и еженедельно.**
12. Команда **subscription-manager** запускает процесс регистрации системы в RHSM.

LAB ОТВЕТЫ

Лаборатория 1

Когда лаборатория будет завершена, выполните следующие команды для проверки соединения:

 # yum clean all

 # yum update

Вывод должен выглядеть примерно так:

Loaded plugins: langpacks, product-id, subscription-manager
inst | 3.7 kB 00:00
inst/primary_db | 2.9 MB 00:00
Setting up Update ProcessНастройка процесса обновления

Эти выходные данные проверяют успешное соединение с сервером FTP. Если вы видите что-то значительно отличается, проверьте следующее в файле **/etc/yum.repos.d/file.repo**:

- Убедитесь, что раздел в этом файле начинается с [**inst**].
- Проверьте **URL-адрес**, связанный с директивой **baseurl**. Он должен соответствовать URL-адресу FTP-сервера определено в главе 1, лабораторная работа 2. Вы должны иметь возможность запускать команды **lftp и ftp с этим URL** из интерфейса командной строки. Если это не работает, либо FTP-сервер не работает, либо сообщения на этот сервер блокируются брандмауэром.
- Если возникли проблемы, устраните их. Затем попробуйте предыдущие команды еще раз.

Лаборатория 2

Один из способов проверить все файлы в каталоге **/usr/sbin - использовать команду rpm -Va | grep /usr/sbin**.

В случае успеха вы обнаружите, что файлы **/usr/sbin/vsftpd и /etc/vsftpd/vsftpd.conf** отличаются от файлов оригинальнычх версий, как установлено из RPM. Изменения в конфигурационном файле не имеют большого значения, особенно если это было настроено любым способом. Тем не менее, изменения в двоичном файле являются причиной подозрений.

Предполагая, что стандартные пакеты Red Hat RPM, удаление и переустановка должны сохранить изменения файл **vsftpd.conf в файле vsftpd.conf.rpmsave.**

Если у вас действительно есть проблемы с безопасностью, необходимы дополнительные меры. Например, некоторые Специалисты по безопасности могут сравнить все файлы в подозрительной системе с файлами в проверенной базовой системе. В этом случае проще всего взять копию или клон базовой системы, переустановить **vsftpd** RPM и перенастройте его по мере необходимости. Предполагая, что базовая система безопасна, вы будете разумно уверены, что новый сервер также будет в безопасности. Изменения, внесенные сценарием в эту лабораторную работу, устанавливают новое время изменения для **/usr/sbin/vsftpd** двоичный файл и добавлен комментарий в конец файла конфигурации vsftpd.conf. Если вы хотите перезагрузить со свежими копиями этих пакетов сделайте резервную копию вашего текущего файла **vsftpd.conf** и запустите **rpm -e vsftpd** Команда для удаления пакета. Если RPM-пакет был перенастроен, вы должны увидеть, по крайней мере, следующее предупреждающее сообщение:

warning: /etc/vsftpd/vsftpd.conf saved as /etc/vsftpd/vsftpd.conf.rpmsave
Затем вы можете переустановить исходный пакет с установочного DVD или из удаленного репозитория.

Кроме того, вы можете удалить (или переместить) измененные файлы, а затем выполнить следующую команду для заставить команду **rpm** предоставить исходные копии этих файлов из соответствующего пакета.
номер версии основан на DVD RHEL 7.0.
 # rpm -ivh --force vsftpd-3.0.2-9.el7.x86_64.rpm

Лаборатория 3

Эта лаборатория предназначена для того, чтобы помочь вам понять, что может делать команда обновления **yum**. Это основной инструментам обновления GUI. Как видно из файла update.txt, созданного в этой лабораторной работе, сообщения отображаются как **yum** отображается для всех новых пакетов из настроенных репозиториев или RHN, загружает их заголовки и использует их для проверки зависимостей, которые также необходимо загрузить и установить.

Лаборатория 4

Эта лаборатория должна быть простой, потому что она предполагает использование инструмента «Настройки обновления программного обеспечения», которую вы можете запустить из командной строки графического интерфейса с помощью команды **gpk-prefs**.

Лаборатория 5

Эта лаборатория не требует пояснений и предназначена для того, чтобы помочь вам понять, что происходит, когда вы правильно установите новое ядро RPM. Как и в других дистрибутивах Linux, при установке (и не использовать режим обновления для) нового ядра, затронуты две области.

Новое ядро добавлено в качестве новой опции в меню конфигурации GRUB2. Существующее ядро должны быть сохранены в качестве опции в этом меню. Когда вы перезагрузите систему, попробуйте новое ядро. Не стесняйтесь перезагрузить систему снова, а затем попробуйте другой вариант, возможно, более старое ядро.

Когда вы просматриваете каталог **/boot**, там должны быть все ранее установленные загрузочные файлы. новое ядро RPM должно добавить соответствующие версии всех одинаковых файлов - с разными номерами ревизий.

Чтобы все было прямо, полезно, если вы сделали копии оригинальных версий GRUB 2. файл конфигурации и список файлов в каталоге **/boot**. Если вы решили сохранить недавно установленный ядро, отлично. В противном случае удалите только что установленное ядро. Это один случай, когда номера ревизий требуются с помощью команды **rpm -e**; следующее основано на удалении ядра и прошивки ядра пакеты, основанные на версии 3.10.0-229.el7:

 # rpm -e kernel-3.10.0-229.el7.x86_64
 # rpm -e linux-firmware-20140911-0.1.git365e80c.el7

Если номер версии ядра или пакета встроенного ПО ядра, который вы установили во время этой лабораторной работы, разные, настройте команды соответственно.

Лаборатория 6

Эта лабораторная работа предназначена для того, чтобы вы могли практиковаться как с командой **yum**, так и с программным обеспечением **Add/Remove Software tool**. Это должно помочь вам подготовиться к главе 9 и предоставить навыки, необходимые для установки служб для других глав. Теперь вы должны понимать, что, поскольку все пакеты в пакете клиентов удаленного рабочего стола группа не обязательна, команда **yum group install “Remote Desktop Clients”** не устанавливает что-нибудь. Вам нужно будет установить каждый из дополнительных пакетов по имени.

Чтобы определить имена устанавливаемых пакетов, запустите информацию группы **yum group info “Remote Desktop Clients”**. Обязательно установите каждый пакет из этой группы в обеих системах. Лучший метод с командой **yum install package1 package2 ...,** где находятся package1, package2 и т. д. имена пакетов в группе пакетов «Клиенты удаленного рабочего стола».