УТВЕРЖДЕН

РСЮК.10201-01 92 01-ЛУ

Операционная система РОСА «КОБАЛЬТ»

Руководство администратора

РСЮК.10201-01 92 01

Листов 303

2018

Литера

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

СОДЕРЖАНИЕ

АННОТАЦИЯ

Настоящий документ является руководством администратора ОС РОСА «КОБАЛЬТ» (далее по тексту — изделие) и содержит условия применения изделия, перечень задач и обязанностей администратора и описание процессов настройки и обслуживания изделия.

1. ВВЕДЕНИЕ

1.1. Аппаратные требования

Для установки и функционирования изделия требуется следующий состав технических средств:

– ЭВМ с процессором архитектуры x86_64;
– не менее 1536 МБ оперативной памяти;
– не менее 10 ГБ свободного места на жестком диске;
– VGA-адаптер и монитор с поддержкой разрешения 1024 × 768 пикс. (24 бит);
– устройство чтения DVD;
– клавиатура;
– мышь.

1.2. Общие принципы работы

Администратор осуществляет установку и настройку ОС в соответствии с настоящим документом.

Администратор, которому предстоит работать с изделием, регистрируется как пользователь. Пользователь имеет учетную запись, для доступа к которой он должен пройти процедуру авторизации. Изделие позволяет одному пользователю авторизоваться в физической консоли электронно-вычислительной машины (далее — ЭВМ). При этом одной физической консоли могут соответствовать несколько виртуальных. Различные пользователи могут одновременно проходить авторизацию с использованием нескольких виртуальных консолей. ЭВМ, работающие под управлением изделия, могут быть соединены между собой физически и логически защищенными локальными сетями.

Конфигурация отдельно стоящей рабочей станции предусматривает возможность использования изделия множеством пользователей. Пользователи имеют возможность работать с изделием после авторизации на терминале.

Сетевая конфигурация изделия предусматривает систему, состоящую из множества ЭВМ, объединенных в сеть, в которой пользователи могут проходить авторизацию на консоли любого из узлов сети. Такая конфигурация является более широкой, чем архитектура одиночной локальной сети, поскольку изделие осуществляет маршрутизацию с использованием протокола IP между различными сегментами сети. Маршрутизация пакетов осуществляется путем их передачи от узла, находящегося в одном сегменте сети, к узлу, находящемуся в другом сегменте, и так далее до достижения целевого узла.

Изделие предоставляет возможность экспорта данных, импорт которых должен поддерживаться всеми участниками взаимодействия. При импорте/экспорте данных необходимо помнить о том, что данные могут быть переданы за пределы контроля ОС (например, при помощи съемных носителей информации), что должно регламентироваться соответствующей политикой безопасности организации.

1.3. Обязанности администратора

К обязанностям администратора относятся:

– обеспечение эффективности работы изделия;
– обеспечение защиты информации от случайного или преднамеренного повреждения;
– управление правами доступа пользователей к функциям изделия.

1.3.1. Обязанности по обслуживанию пользователей

Обязанности по обслуживанию пользователей:

– предоставление пользователям доступа к ресурсам;
– оценка потребностей пользователей;
– планирование дальнейшего развития/изменения системы;
– оказание помощи пользователям.

1.3.2. Обязанности по поддержке оборудования

Обязанности по поддержке оборудования:

– проверка корректности установки периферийных устройств;
– обеспечение надлежащей производительности аппаратного обеспечения;
– восстановление работоспособности в случае отказа оборудования.

1.3.3. Обязанности по управлению программным обеспечением

Обязанности по управлению программным обеспечением (далее — ПО):

– создание файловой системы (далее — ФС);
– управление целостностью ФС;
– контроль использования ресурсов ЭВМ;
– поддержка процедур резервного копирования и восстановления изделия и данных;
– обновление изделия;
– установка и обновление ПО.

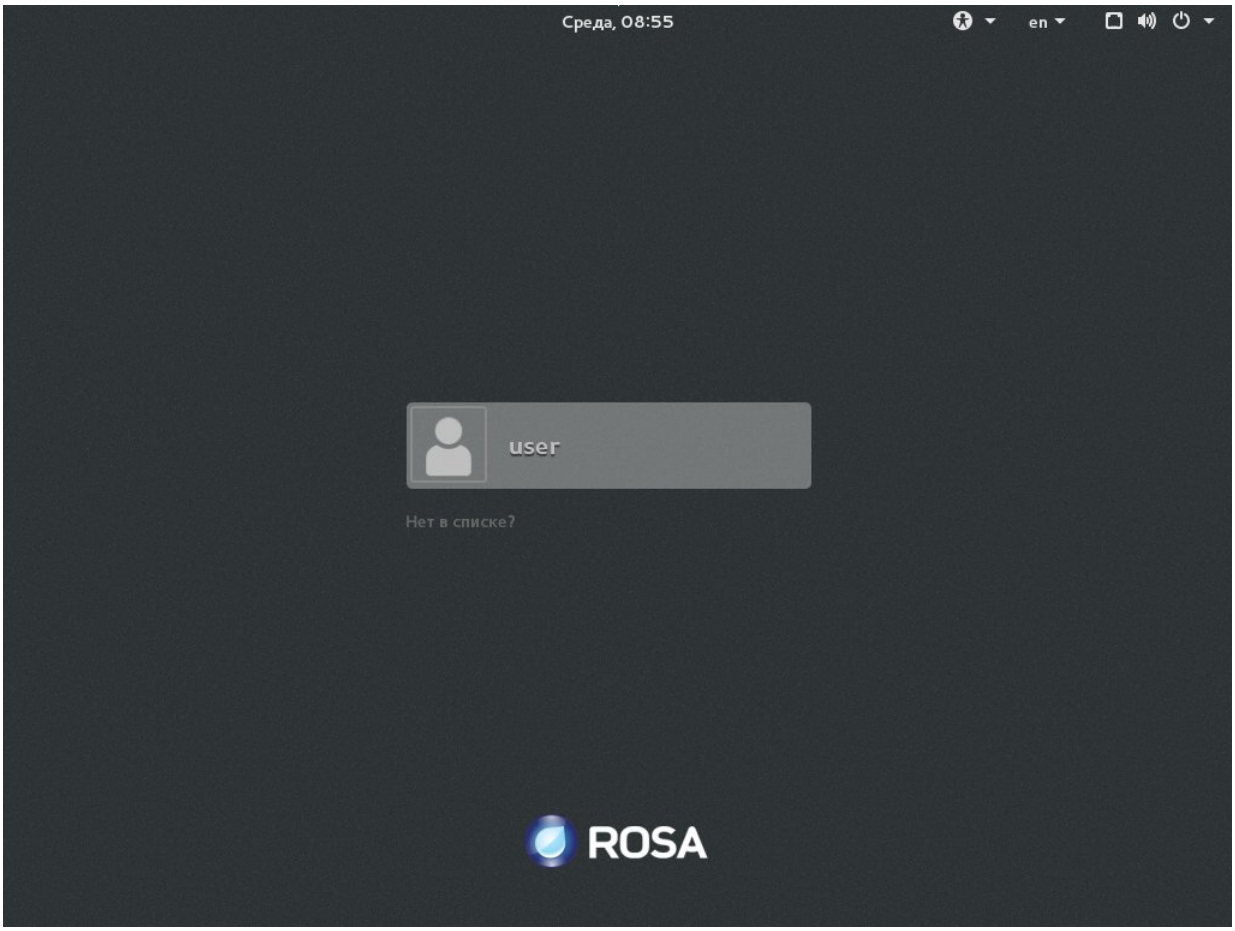
1.4. События безопасности и сообщения

В процессе работы с ОС администратор может получать сообщения:

– о критических событиях безопасности;
– об ограничении ресурсов;
– о некорректном использовании ПО или параметров, обрабатываемых ПО, и др.

2. БАЗОВАЯ НАСТРОЙКА ПОСЛЕ УСТАНОВКИ ДЛЯ РАБОЧИХ СТАНЦИЙ

Перед выполнением любых действий в системе необходимо авторизоваться. Если была выполнена установка ОС с набором пакетов по умолчанию, для входа будет использоваться соответствующий экран графической оболочки MATE:



Экран входа MATE

Если был выбран минимальный вариант установки ОС, после перезагрузки вы увидите строку приглашения командного интерпретатора:

Строка приглашения в консольном режиме

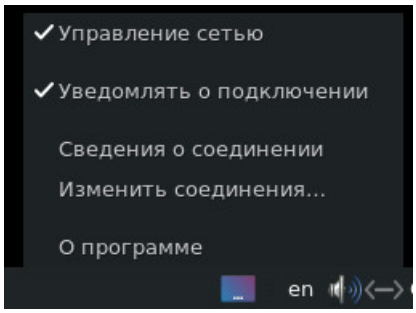
Вне зависимости от того, какой способ входа был выбран, вам необходимо ввести имя учетной записи пользователя и пароль.

Также вы всегда можете переключиться между графическим и консольным интерфейсом, использовав комбинацию клавиш <Ctrl+Alt+FN>, где N — номер виртуальной консоли. По умолчанию графическому интерфейсу соответствует клавиша <F1>.

2.1. Настройка сетевых соединений

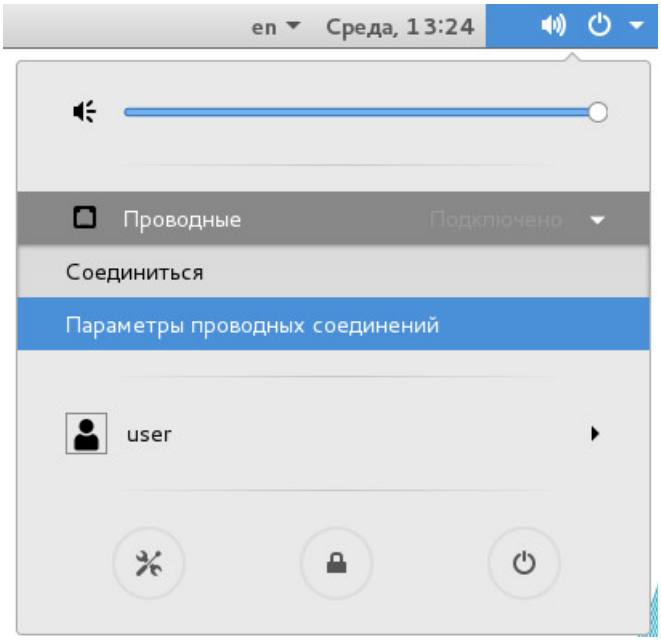
Примечание. Для внесения изменений в сетевые настройки потребуются привилегии администратора системы (пользователя, входящего в группу wheel) или суперпользователя root.

Для среды MATE в правом нижнем углу щелкните правой кнопкой мыши по значку сетевого соединения. Внешний вид значка может отличаться в зависимости от того, присутствует ли в системе беспроводной модуль связи Wi-Fi. Если он есть, будет показан стандартный знак Wi-Fi, если нет — знак примет следующий вид: ↔. Выберите пункт «Изменить соединения...».



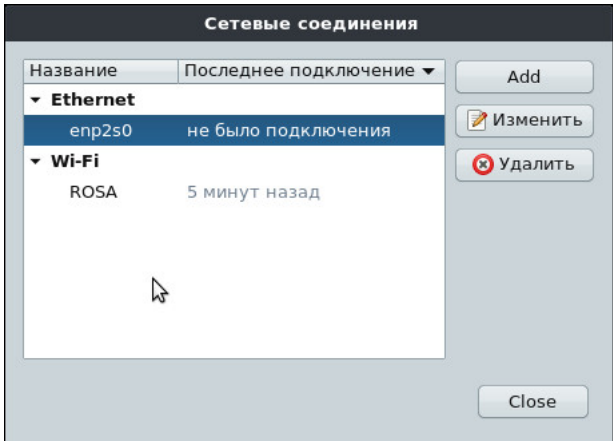
Управление сетью MATE

В случае GNOME: стрелка в правом верхнем углу → значок «сеть» → нужный вариант сети (проводная, беспроводная). Выберите пункт «Параметры соединений».



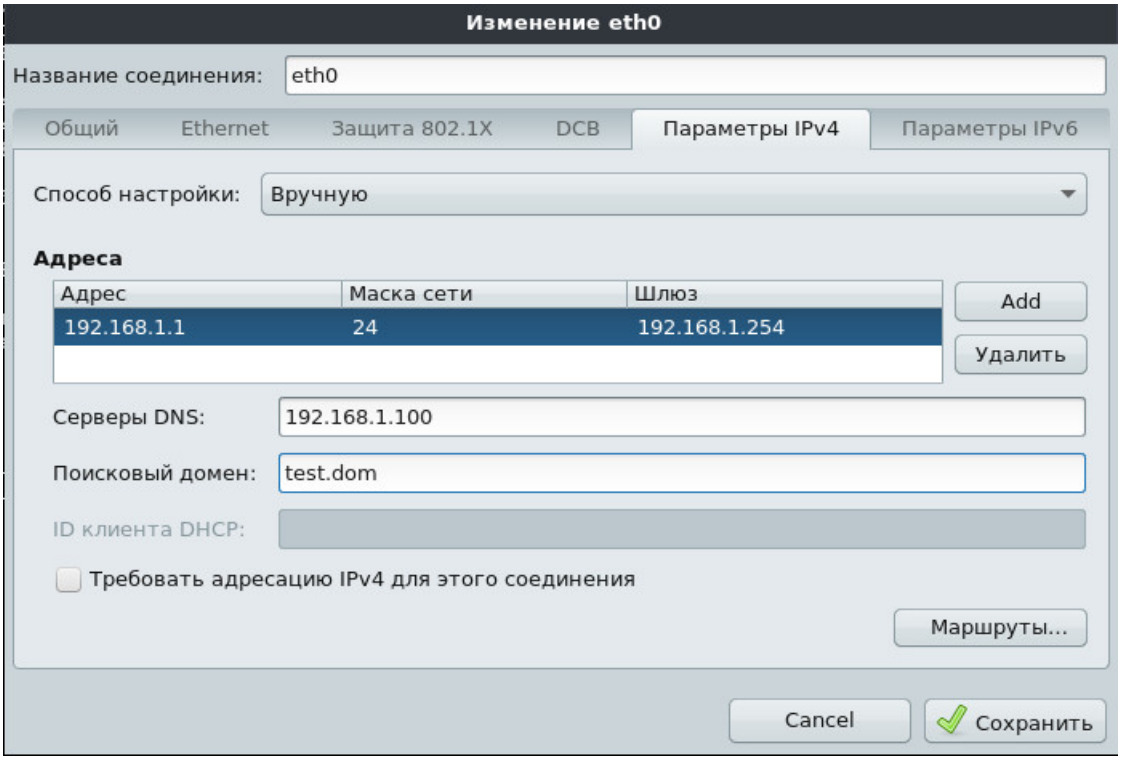
Управление сетью GNOME

В появившемся окне «Сетевые соединения» (MATE) или «Сеть» (GNOME) можно изменять параметры существующих сетевых соединений и добавлять новые.



Сетевые соединения MATE

Поддерживается создание как физических (проводных/беспроводных), так и логических (VPN, VLAN, BOND, мост) соединений. В зависимости от типа выбранного соединения будут доступны доступны различные параметры, поддерживаемые данным типом соединения. Например, для проводного варианта набор возможных настроек будет выглядеть, как показано на рисунке:



Настройка проводного соединения

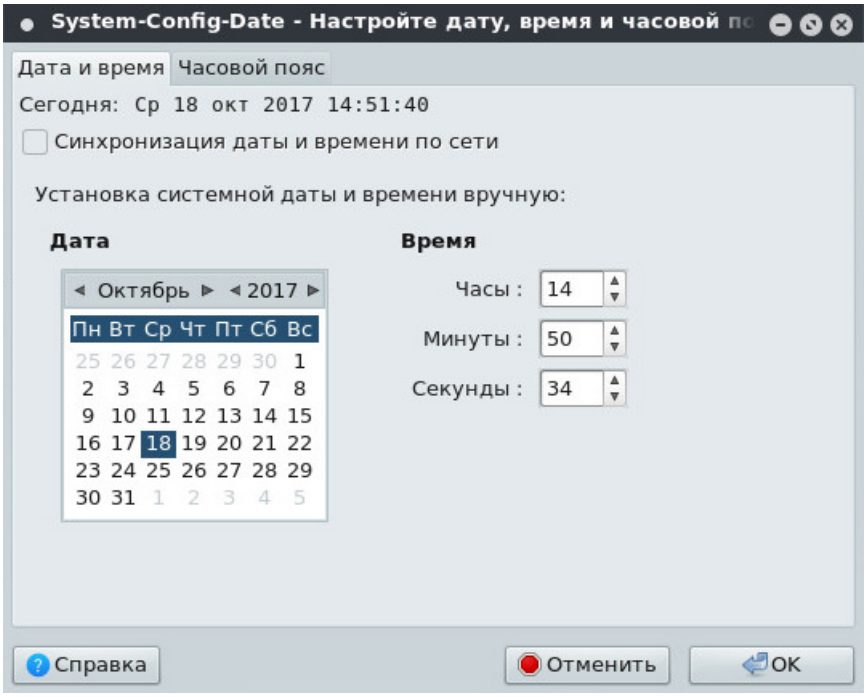
После завершения настройки нажмите на кнопку [Сохранить]. Параметры нового соединения будут определены автоматически.

2.2. Настройка времени системы

Примечание. Для внесения изменений в настройки даты и времени потребуются привилегии администратора системы (пользователя, входящего в группу wheel) или суперпользователя root.

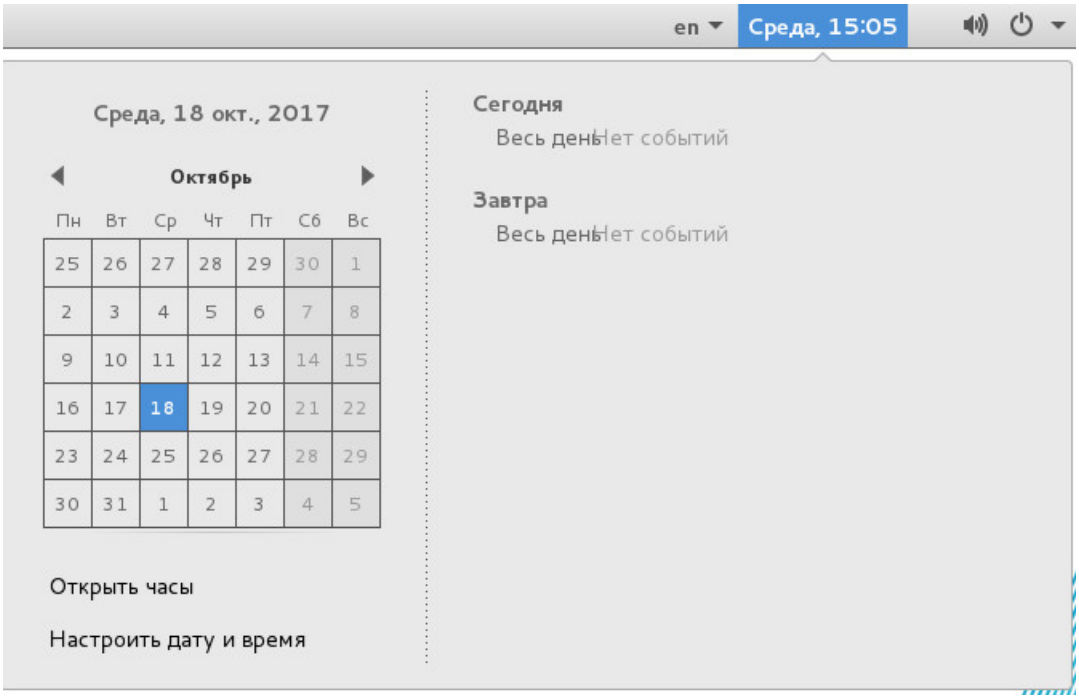
Для настройки времени в среде MATE используйте утилиту *System-Config-Date*, расположенную по пути «Система → Инструменты для администратора → System-Config-Date». Ее также можно запустить командой `system-config-date`.

Чтобы задать дату и время вручную, снимите галочку с пункта «Синхронизация даты и времени по сети».

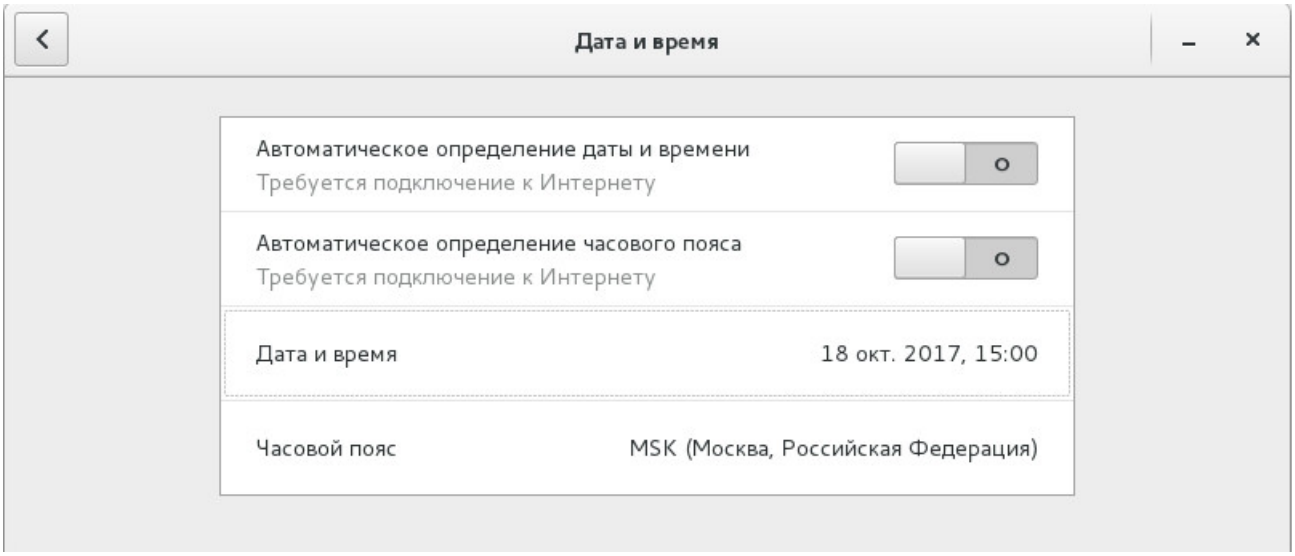


Настройка даты и времени MATE

Для настройки времени в среде GNOME используйте утилиту «Дата и время». Для этого выберите в верхнем правом углу текущее время и в открывшемся окне нажмите на кнопку [Настроить дату и время].



Настройка даты и времени GNOME



Параметры даты и времени GNOME

2.3. Установка пароля на загрузчик

Для установки пароля на загрузчик Grub2 необходимо выполнить следующие действия:

1) Войти в систему под учетной записью суперпользователя root или повысить привилегии до его уровня командой `su` / `sudo`.
2) Сгенерировать хеш пароля:
   ```
   # grub2-mkpasswd-pbkdf2
   ```
3) Добавить в файл /etc/grub.d/40_custom следующие строки:
   ```
   set superusers="root"
   password_pbkdf2 root <хеш>
   ```
   Вместо `<хеш>` нужно указать сгенерированный хеш пароля.
4) Обновить основной файл настройки загрузчика /boot/grub2/grub.cfg:
   ```
   # grub2-mkconfig -o /boot/grub2/grub.cfg
   ```
   Если загрузчик работает в режиме UEFI, расположение последнего конфигурационного файла будет иным. Обычно он находится в каталоге /boot/EFI/grub2/.
5) Перезагрузить ОС.

2.4. Настройка экрана входа в систему

Чтобы отключить отображение списка пользователей в графическом интерфейсе входа в систему, необходимо:

1) Создать файл /etc/dconf/profile/gdm со следующим содержанием:
   ```
   user-db:user
   system-db:gdm
   file-db:/usr/share/gdm/greeter-dconf-defaults
   ```
2) Создать файл /etc/dconf/db/gdm.d/00-login-screen со следующим содержанием:
   ```
   [org/gnome/login-screen]
   ```

```
# Do not show the user list
disable-user-list=true
```

3) Обновить систему конфигурации *dconf* следующей командой:

```
# dconf update
```

4) Перезапустить службу дисплейного менеджера:

```
# service gdm restart
```

3. УСТАНОВКА ПРИЛОЖЕНИЙ. МЕНЕДЖЕР ПАКЕТОВ YUM

Yum — это менеджер пакетов в ОС РОСА «КОБАЛЬТ», который может выполнять запросы информации о доступных пакетах, загружать пакеты из репозиториев, устанавливать и удалять их из системы, а также обновлять всю систему до последней доступной версии. Во время обновления, установки или удаления пакетов *yum* автоматически выполняет разрешение зависимостей, т. е. определяет, скачивает и устанавливает все доступные пакеты зависимостей.

Для работы *yum* можно подключать новые дополнительные репозитории (источники пакетов), а также множество модулей, расширяющих и дополняющих его возможности. Задачи, выполняемые yum, аналогичны задачам, которые выполняет пакетный менеджер RPM; многие из консольных команд этих пакетных менеджеров идентичны. *Yum* предоставляет возможность легкого управления пакетами на одной машине или в группе машин.

Примечания.

1. ОС РОСА «КОБАЛЬТ» — сертифицированный дистрибутив, предназначенный для работы в защищенных системах. По умолчанию используется только один репозиторий (DVD), файлы которого прошли проверку на отсутствие недекларированных возможностей (НДВ), и установка программных средств из других источников может скомпрометировать вашу систему. Установку сторонних программ нужно согласовывать с администратором безопасности информации и органом, выдавшим аттестат на использование вашей защищенной системы.

Если ОС РОСА «КОБАЛЬТ» не планируется использовать в защищенных системах, ограничения на использования сторонних репозиториев не накладываются.

2. *Yum* предоставляет защищенное управление пакетами с помощью проверки подписи GPG (инструмента для безопасных коммуникаций GNU Privacy Guard; также известен как GNUPG) на подписанных пакетах. Проверку подписей GPG можно включить для всех или для отдельных репозиториев. При включенной возможности проверки подписей *yum* откажется устанавливать пакеты, у которых отсутствует подпись GPG с корректным ключом. Это означает, что пользователь сможет загружать и устанавливать пакеты RPM только из доверенных источников, и эти пакеты гарантированно не были изменены во время передачи.

3. Чтобы *yum* смог установить, обновить или удалить пакеты в системе, необходимо обладать правами суперпользователя root. Все примеры в этом разделе предполагают, что эти права уже получены.

3.1. Обновление пакетов

Yum способен проверить наличие обновлений для пакетов, установленных в системе. Пользователь может указать список пакетов и обновить их все сразу или по отдельности.

Примечание. Если ОС РОСА «КОБАЛЬТ» планируется использовать в защищенных системах, обновление пакетов возможно только с официальных версий дистрибутива или обновлений к нему, прошедших инспекционный контроль.

3.1.1. Проверка наличия обновлений

Чтобы просмотреть, для каких из установленных в системе пакетов имеются обновления, выполните следующую команду:

```
# yum check-update
```

Пользователь может выбрать обновление одного пакета, нескольких пакетов или всех пакетов сразу. Если и для зависимостей обновляемых пакетов имеются обновления, они также будут установлены.

3.1.2. Обновление одного пакета

Чтобы обновить один пакет, выполните следующую команду:

```
# yum update <имя_пакета>
```

Yum предоставляет информацию об обновлении и затем предлагает пользователю его подтвердить. По умолчанию *yum* работает в интерактивном режиме. Если пользователь уже знает, какие действия *yum* планирует выполнить, можно указать параметр `-y` для автоматического ответа на вопросы, задаваемые yum (в этом случае он будет выполняться неинтерактивно). Тем не менее, всегда рекомендуется просмотреть, какие изменения *yum* планирует внести в систему, чтобы не оказаться безоружным при возникшей проблеме. Также можно выбрать возможность простой загрузки пакетов без их установки. Для этого в предложении загрузки введите `d`. Это запустит фоновую загрузку выбранного пакета.

Если транзакция окончилась неудачей, можно просмотреть историю транзакций *yum* с помощью команды `yum history`.

Примечание. *Yum* всегда устанавливает новое ядро вне зависимости от того, была ли указана команда `yum update` или `yum install`. С другой стороны, при использовании RPM важно использовать команду `rpm -i kernel`, устанавливающую новое ядро, вместо команды `rpm -u kernel`, которая заменяет текущее ядро.

Точно так же можно обновить и группу пакетов:

```
# yum group update <имя_группы_пакетов>
```

Замените `<имя_группы_пакетов>` на имя группы пакетов, которую нужно обновить.

3.1.3. Обновление всех пакетов в системе и их зависимостей

Чтобы обновить все пакеты и их зависимости, выполните следующую команду:

```
# yum update
```

Аргументы не требуются.

3.2. Поиск пакетов

Пользователь может выполнять поиск по имени пакета RPM, описанию и сводке с помощью следующей команды:

```
$ yum search <имя_пакета>
```

3.2.1. Поиск пакетов по соответствию строке

Чтобы получить список всех пакетов, соответствующих строкам «vim», «gvim» или «emacs», выполните:

```
$ yum search vim gvim emacs
```

Команда `yum search` удобна для поиска пакетов, точное имя которых пользователь не знает, но для которых известен связанный с ними термин. Обратите внимание, что по умолчанию поиск yum возвращает соответствия и в имени пакета и в сводке, что ускоряет поиск. Для более подробного, но чуть более медленного поиска используйте команду `yum search all`.

3.2.2. Фильтрация результатов поиска

Все команды поиска *yum* предоставляют пользователю возможность фильтрации результата с помощью добавления одного или более шаблонов выражений в качестве аргумента. Шаблоны выражений — это обычные строки символов, содержащие один или несколько символов подстановки «*» (который расширяется до соответствия любому поднабору знаков) и символа «?» (который расширяется до соответствия любому одиночному символу).

Не забывайте об экранировании шаблонов выражений, указывая их в качестве аргументов для команды *yum*. В противном случае командный интерпретатор Bash обработает эти выражения как расширения имени пути и может передать *yum* все файлы в текущем каталоге, совпадающие с шаблоном. Чтобы корректно передать все шаблоны выражений *yum*, используйте один из следующих приемов:

1) Экранируйте символы подстановки, поставив перед ними символ косой черты.
2) Заключите все выражение-шаблон в одинарные или двойные кавычки.

Нижеприведенные примеры показывают использование обоих этих способов.

Пример 1. Вывод списка пакетов

Чтобы получить список всех установленных и доступных к установке пакетов, выполните следующую команду:

```
$ yum list all
```

Чтобы получить список установленных и доступных к установке пакетов, соответствующих указанному шаблону выражения, выполните:

```
$ yum list <шаблон_выражения>
```

Пример 2. Вывод списка всех пакетов, имеющих отношение к ABRT

Пакеты, содержащие различные добавления и модули ABRT, начинаются либо с «abrt-addon-», либо с «abrt-plugin-». Чтобы получить список этих пакетов, выполните следующую команду:

```
$ yum list abrt-addon\* abrt-plugin\*
```

Обратите внимание, что символы подстановки экранируются косой чертой.

Чтобы получить список всех пакетов, установленных в системе, используйте ключевое слово «installed». Крайний правый столбец в выводе показывает список репозиториев, из которых пакеты были получены.

```
$ yum list installed <шаблон_выражения>
```

Пример 3. Получение списка всех установленных версий пакета krb

В примере ниже показывается, как получить список всех установленных пакетов, в названии которых сначала идет комбинация «krb», затем ровно один символ и дефис. Это удобно, если пользователь хочет получить список всех версий какого-то одного компонента, различающихся по номерам. Для корректной обработки запроса весь шаблон выражения заключен в кавычки.

```
$ yum list installed "krb?-*"
```

Чтобы получить список всех доступных для установки пакетов во всех подключенных репозиториях, используйте команду в следующем виде:

```
$ yum list available <шаблон_выражения>
```

Пример 4. Получение списка всех доступных модулей gstreamer

Чтобы получить, например, список всех доступных пакетов, имена которых которых в начале содержат «gstreamer», а затем «plugin» (модуль), выполните следующую команду:

```
$ yum list available gstreamer\*plugin\*
```

3.3. Получение списка репозиториев

Чтобы получить список идентификаторов репозиториев, имя репозитория и число пакетов в каждом активном репозитории в системе, выполните следующую команду:

```
$ yum repolist
```

Чтобы получить дополнительную информацию об этих репозиториях, добавьте параметр `-v`. Для каждого репозитория в списке будет показана информация об имени файла, общем размере, дате подледного обновления и основном адресе URL. Как вариант, можно использовать команду `repoinfo`, которая выдаст такой же результат.

```
$ yum repolist -v
$ yum repoinfo
```

Чтобы получить список как подключенных, так и неподключенных репозиториев, используйте следующую команду с аргументом `all`. В список добавится столбец со статусом репозитория, в котором будет указано, какие из репозиториев активны, а какие — нет.

```
$ yum repolist all
```

Передав в качестве первого аргумента `disabled`, можно сузить вывод до неподключенных репозиториев. Для еще большей конкретизации в качестве аргументов можно передать идентификатор или имя репозитория, а также связанный шаблон выражения. Обратите внимание, что в случае точного совпадения ID или имени репозитория и указанных аргументов репозиторий будет показан, даже если он не соответствует фильтру «подключен/не подключен».

3.4. Получение информации о пакетах

Чтобы получить информацию об одном или нескольких пакетах, выполните следующую команду (здесь также можно применять шаблоны выражений):

```
$ yum info <имя_пакета>
```

Чтобы узнать сведения о пакете mc, выполните:

```
$ yum info mc
```

Команда `yum info <имя_пакета>` аналогична команде `rpm -q --info <имя_пакета>`, но в качестве дополнительной информации предоставляет имя репозитория, из которого был установлен пакет RPM.

3.5. Работа с yumdb

Чтобы получить альтернативную и полезную информацию о пакете, пользователь также может сделать запрос в базе данных yum с помощью следующей команды:

```
# yumdb info имя_пакета
```

Эта команда предоставляет дополнительную информацию о пакете:

- контрольную сумму (а также используемый для ее вычисления алгоритм, например, SHA-256);
- команду, с помощью которой был установлен пакет (при ее наличии);
- причину, по которой пакет установлен в системе.

3.6. Установка пакетов

Чтобы установить один пакет и все его неустановленные зависимости, выполните следующую команду с привилегиями суперпользователя root:

```
# yum install <имя_пакета>
```

Также можно установить одновременно несколько пакетов, добавив их имена в качестве аргументов. Для этого выполните:

```
# yum install <имя_пакета> <имя_пакета> ...
```

При установке пакетов в системе multilib, такой, как AMD64 или Intel 64, можно указать архитектуру пакета (если она доступна в подключенном репозитории), добавив к имени пакета «.arch».

```
# yum install <имя_пакета.arch>
```

3.6.1. Установка пакетов в системе multilib

Чтобы установить пакет sqlite для архитектуры i686, выполните следующую команду с привилегиями суперпользователя root:

```
# yum install sqlite.i686
```

Для быстрой установки нескольких пакетов с похожими именами можно использовать шаблоны выражений. Выполните:

```
# yum install <шаблон_выражения> ...
```

3.6.2. Установка всех модулей audacious

Шаблоны выражений удобны, если нужно установить несколько пактов с похожими именами. Чтобы установить все модули audacious, используйте команду в таком виде:

```
# yum install audacious-plugins-\*
```

В дополнение к именам пакетов и шаблонам выражений, команде `yum install` также можно передавать имена файлов. Если известно имя выполняемого файла пакета,

но не само имя пакета, то команде `yum install` можно передать имя пути. Выполните:

```
# yum install /usr/sbin/named
Yum then searches through its package lists, finds the package
which provides /usr/sbin/named, if any, and prompts you as to
whether you want to install it.
```

Yum выполнит поиск по списку пакетов, найдет пакет, который предоставляет файл /usr/sbin/named, если такой есть, и выведет запрос подтверждения установки пакета.

Как можно видеть в вышеприведенных примерах, команде `yum install` не требуются четкие аргументы. Она может обрабатывать различные форматы имен пакетов и шаблонов выражений, что облегчает пользователям установку. С другой стороны, на корректную обработку команды `yum` требуется время, особенно если было указано большое число пакетов. Для оптимизации поиска пакетов можно использовать следующие команды, явным образом указывающие, как именно необходимо обрабатывать аргументы:

```
yum install-n <имя>
yum install-na <имя.архитектура>
yum install-nevra <имя-epoch:версия-релиз.архитектура>
```

При использовании аргумента `install-n` команда `yum` воспринимает имя как точное имя пакета. Команда `install-na` указывает `yum`, что последующий аргумент содержит имя пакета и архитектуру, разделенные символом точки. С аргументом `install-nevra` команда `yum` ожидает аргумента в виде `<имя-epoch:версия-релиз.архитектура>`. Точно так же при поиске пакетов для удаления можно использовать `yum remove-n`, `yum remove-na` и `yum remove-nevra`.

Примечание. Если необходимо установить пакет с именованным бинарным файлом, но неизвестно, в какой каталог — bin/ или sbin/ — он устанавливается, используйте команду `yum provides` с шаблоном подстановки:

```
# yum provides "*bin/named"
```

3.7. Загрузка пакетов

На определенном моменте установки пользователю выводится запрос о подтверждении установки со следующим сообщением:

```
...
Total size: 1.2 M
Is this ok [y/d/N]:
...
```

При использовании с ключом `d` команда `yum` только загружает пакеты и не устанавливает их немедленно. Эти пакеты можно установить позже, с помощью команды `yum localinstall`, или же их можно сделать их общими с помощью какого-либо другого устройства. Загруженные пакеты хранятся в одном из подкаталогов каталога cache, по умолчанию это /var/cache/yum/x86_64/7Server/packages/.

Загрузка выполняется в фоновом режиме, так что пользователь может одновременно использовать *yum* для параллельных задач.

3.8. Удаление пакетов

Yum предоставляет пользователю как средство установки пакетов, так и средство их удаления. Чтобы удалить конкретный пакет, а также пакеты, зависящие от этого пакета, выполните с привилегиями суперпользователя root:

```
yum remove <имя_пакета>...
```

Так же, как и при установке нескольких пакетов, пользователь может удалить несколько пакетов, указав их имена в команде.

Аналогично команде `install`, команда `remove` может принимать следующие аргументы:

- имена пакетов;
- шаблоны выражений;
- список файлов;
- поставщик пакетов.

Примечание. *Yum* не может удалить пакет, не удалив также и пакеты, зависящие от него. Такие действия, которые может выполнить только пакетный менеджер RPM, не рекомендуются, и могут привести систему в нерабочее состояние, или же стать причиной некорректной работы или внезапных отказов в работе приложений.

3.9. Работа с историей транзакций yum

Команда `yum history` дает пользователю возможность просмотреть информацию о выполненных командах `yum`, о датах и времени их выполнения, о числе затронутых пакетов, о том, были ли эти транзакции успешными или же были прерваны, и была ли изменена база данных RPM в промежуток между транзакциями. Кроме того, с помощью этой команды можно повторить или отменить некоторые транзакции. Все данные истории хранится в базе данных истории в каталоге /var/lib/yum/history/.

3.9.1. Получение списка транзакций

Чтобы получить список двадцати последних транзакций, выполните с привилегиями суперпользователя root команду `yum history` без дополнительных аргументов либо с аргументом `list`:

```
# yum history list
```

Чтобы увидеть все транзакции, добавьте ключевое слово `all`:

```
# yum history list all
```

Чтобы увидеть транзакции, выполненные в определенный промежуток времени, выполните команду в следующем виде:

```
# yum history list start_id..end_id
```

Также можно просмотреть транзакции для конкретного пакета или пакетов. Для этого к команде добавляется имя пакета или шаблон выражения:

```
# yum history list <шаблон_выражения>...
```

3.9.2. Вывод списка пяти последних транзакций

В выводе команды `yum history list` недавние транзакции показываются в верхней части списка. Чтобы увидеть информацию о пяти самых старых транзакциях в базе данных истории, выполните:

```
# yum history list 1..5
```

Все формы команды `yum history list` предоставляют вывод в виде таблицы, где каждая строка состоит из следующих столбцов:

- `ID` — целое значение, обозначающее конкретную транзакцию;
- `Login user` — имя пользователя, в сеансе работы которого была произведена транзакция. Эта информация предоставляется в виде «полное имя <имя_пользователя>». Для транзакций, произведенных не пользователями (например, автоматическое обновление системы) вместо этого используется System <unset>;
- `Date and time` — дата и время начала транзакции;
- `Action(s)` — список действий, которые были выполнены во время транзакции, согласно списку «Возможные значения поля Action(s)», приведенному ниже;
- `Altered` — количество пакетов, затронутых транзакцией. Далее может указываться дополнительная информация, согласно списку «Возможные значения поля Altered», приведенному ниже.

3.9.2.1. Возможные значения поля Action(s)

- `Downgrade (D)` — как минимум один пакет был понижен до более старой версии;
- `Erase (E)` — как минимум один пакет был удален;
- `Install (I)` — как минимум один новый пакет был установлен;
- `Obsoleting (O)` — как минимум один пакет был помечен как устаревший;
- `Reinstall (R)` — как минимум один пакет был переустановлен;
- `Update (U)` — как минимум один пакет был обновлен но свежей версии.

3.9.2.2. Возможные значения поля Altered

- `<` — перед началом транзакции база данных rpmdb была изменена за пределами *yum*;
- `>` — после окончания транзакции база данных rpmdb была изменена за пределами *yum*;
- `*` — транзакцию не удалось завершить;
- `#` — транзакция была завершена успешно, но *yum* вернул ненулевое значение на выходе;
- `E` — транзакция была завершена успешно, но была выведена ошибка или предупреждение;
- `P` — транзакция была завершена успешно, но в базе данных rpmdb уже существовала ошибка;
- `s` — транзакция была завершена успешно, но был использован аргумент `--skip-broken`, и некоторые пакеты были пропущены.

Чтобы синхронизировать содержимое базы данных rpmdb или yumdb любого уставленного на данный момент пакета с текущей используемой базой данных rpmdb или yumdb, выполните следующую команду:

```
# yum history sync
```

Чтобы просмотреть некоторую общую статистику о текущей используемой базе данных истории транзакций, выполните следующую команду:

```
# yum history stats
```

Yum также дает возможность просмотреть сводку всех прошлых транзакций. Для этого выполните:

```
# yum history summary
```

Чтобы просмотреть все транзакции за указанный период, выполните:

```
# yum history summary start_id..end_id
```

По аналогии с командой `yum history list`, также можно просмотреть историю транзакции одного конкретного пакета или пакетов, указав имя пакета или шаблон выражения:

```
# yum history summary glob_expression ...
```

Все формы команды `yum history summary` создают упрощенный табличный вывод, аналогичный выводу команды `yum history list`. Как показывалось выше, обе команды, `yum history list` и `yum history summary`, направлены на транзакции, и хотя они и дают возможность просмотреть транзакции для одного или нескольких указанных пакетов, в их выводе отсутствуют важные детали, такие, как версии пакетов. Чтобы получить список транзакций с точки зрения пакета, выполните с привилегиями суперпользователя root:

```
# yum history package-list glob_expression ...
```

3.9.3. Отслеживание истории одного пакета

Чтобы, например, отследить историю subscription-manager и связанных с ним пакетов, выполните следующую команду с привилегиями суперпользователя root:

```
# yum history package-list subscription-manager\*
```

Чтобы просмотреть сводку одной транзакции под учетной записью root, используйте команду в следующем виде:

```
# yum history summary id
```

Здесь `id` означает идентификатор транзакции.

Чтобы проверить какую-то конкретную транзакцию или транзакции с большими подробностями, выполните:

```
# yum history info id ...
```

Аргумент `id` опционален, и при его отсутствии yum автоматически использует самую последнюю транзакцию. Обратите внимание, что при указании более одной транзакции также можно использовать диапазон:

```
# yum history info start_id..end_id
```

Также пользователь может просматривать дополнительную информацию, например, какие значения параметров конфигурации использовались во время транзакции, или

из какого репозитория и почему были установлены конкретные пакеты. Чтобы определить, какой тип дополнительной информации доступен для конкретной транзакции, выполните:

```
# yum history addon-info id
```

По аналогии с командой `yum history info`, при отсутствии `id` *yum* автоматически использует самую последнюю транзакцию. Еще одним способом обратиться к самой последней транзакции является использование ключевого слова `last`:

```
# yum history addon-info last
```

В выводе команды `yum history addon-info` содержится следующая информация:

- `config-main` — глобальные параметры yum, которые использовались во время этой транзакции;
- `config-repos` — параметры для отдельных репозиториев yum;
- `saved_tx` — данные, которые может использовать команда `yum load-transaction` для воспроизведения транзакции на другой машине.

Чтобы просмотреть выбранный тип дополнительной информации, выполните:

```
# yum history addon-info id information
```

3.9.4. Откат и повторение транзакций

Помимо проверки истории транзакций, команда `yum history` может откатить или повторить выбранную транзакцию. Чтобы откатить транзакцию, ведите с привилегиями суперпользователя root:

```
# yum history undo id
```

Чтобы повторить конкретную транзакцию, выполните:

```
# yum history redo id
```

Обратите внимание, что обе команды отменяют или повторяют только те шаги, которые были выполнены во время транзакции. Если во время транзакции был установлен новый пакет, команда `yum history undo` его удалит, а если транзакция удалила пакет, команда вновь его установит. Эта команда также будет пытаться вернуть предыдущую версию обновленных пакетов, если эти более старые пакеты все еще доступны.

При управлении несколькими идентичными системами *yum* также дает возможность выполнить транзакцию на одной из них, сохранить подробности транзакции в файле, и после проверки повторить ту же самую транзакцию. Также и на оставшихся системах. Чтобы сохранить детали транзакции в файле, выполните:

```
# yum -q history addon-info id saved_tx > <имя_файла>
```

Сразу после копирования файла на целевую систему транзакцию можно повторить, выполнив следующую команду:

```
# yum load-transaction <имя_файла>
```

Команду `load-transaction` можно настроить так, чтобы она игнорировала недостающие пакты или версию rpmdb. Подробности об этих параметрах см. на странице руководства yum.conf(5).

3.9.5. Создание новой истории транзакций

Yum хранит историю транзакций в одном файле базы данных SQLite. Чтобы начать новую историю транзакций, выполните следующую команду:

```
# yum history new
```

Это действие создаст новую, пустую базу данных в каталоге /var/lib/yum/history/. Старая история транзакций будет сохранена, но закрыта для доступа, пока более новый файл базы данных будет присутствовать в каталоге.

3.10. Настройка параметров и репозиториев yum

Информация о параметрах *yum* и связанных с ним утилит хранится в файле /etc/yum.conf. В этом файле содержится один обязательный раздел `[main]`, в котором настраиваются глобальные параметры *yum*, а также могут содержаться один или более разделов `[repository]`, в которых настраиваются параметры, имеющие отношение к репозиториям. Тем не менее, настраивать индивидуальные репозитории рекомендуется в новых или уже существующих файлах .repo в каталоге /etc/yum.repos.d/. Параметры, определенные в индивидуальных разделах `[repository]` файла /etc/yum.conf, имеют приоритет над параметрами, присутствующими в разделе `[main]`.

В данном подразделе рассказывается о том, как:

- установить глобальные параметры *yum* с помощью редактирования раздела `[main]` файла конфигурации /etc/yum.conf;
- установить параметры индивидуальных репозиториев с помощью редактирования раздела `[repository]` в файле /etc/yum.conf и файлов .repo в каталоге /etc/yum.repos.d/;
- использовать переменные *yum* в файле /etc/yum.conf и файлах в каталоге /etc/yum.repos.d/ так, чтобы значения динамической версии обрабатывались корректно;
- добавлять, подключать и отключать репозитории *yum* в командной строке;
- настроить собственный пользовательский репозиторий *yum*.

3.10.1. Параметры раздела [main]

В конфигурационном файле /etc/yum.conf содержится только один раздел `[main]`, и в то время, как одни пары «ключ-значение» в этом разделе влияют на то, как действует *yum*, другие влияют на то, как *yum* обрабатывает репозитории. Под заголовком `[main]` в файле /etc/yum.conf можно добавить много дополнительных параметров.

Пример файла конфигурации /etc/yum.conf:

```
[main]
[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

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

Наиболее часто используемые параметры в разделе `[main]`:

`assumeyes=<значение>`

Параметр `assumeyes` определяет, будет ли *yum* выводить запрос о подтверждении критически важных действий. Замените `<значение>` одним из следующих:

- 0 (значение по умолчанию) — *yum* выводит запрос о подтверждении выполняемых им критически важных действий;
- 1 — не выводить запрос о подтверждении критически важных действий. При установленном `assumeyes=1` *yum* ведет себя так же, как и с консольными параметрами `-y` и `--assumeyes`.

`cachedir=<каталог>`

Используйте этот параметр для указания каталога, в котором *yum* хранит свой кэш и файлы базы данных. Замените `<каталог>` абсолютным путем до каталога. По умолчанию каталогом кэша *yum* является /var/cache/yum/$basearch/$releasever/.

`debuglevel=<значение>`

Этот параметр определяет степень детализации отладочного вывода *yum*. Здесь `<значение>` — это целое число в диапазоне от 1 до 10. Более высокое значение для `debuglevel` означает более высокий уровень детализации. Значение `debuglevel=2` выставляется по умолчанию, а `debuglevel=0` отключает отладочный вывод.

`exactarch=<значение>`

Этот параметр заставляет *yum* принимать во внимание точное значение архитектуры при обновлении уже установленных пакетов. Замените `<значение>` на одно из следующих:

- 0 — при обновлении пакетов не принимать во внимание точное значение архитектуры;
- 1 (по умолчанию) — учитывать точное значение архитектуры при обновлении пакетов. С этим параметром *yum* не станет устанавливать 32-битный пакет для обновления уже установленного 64-битного пакета.

`exclude=<имя_пакета> [<еще_имена_пакетов>]`

Параметр `exclude` дает возможность исключать пакеты по ключевому слову во время установки или обновления системы. Указать множество пакетов для исключения можно с помощью заключения в кавычки списка пакетов, разделенных пробелами. Также можно использовать шаблоны выражений с символами подстановки (например, `*` и `?`).

gpgcheck=<значение>

Параметр gpgcheck используется для указания, должен ли *yum* выполнять проверку подписи GPG в пакетах. Замените <значение> на одно из следующих:

- 0 — отключить проверку подписи для пакетов во всех репозиториях, включая локальную установку пакетов;
- 1 (по умолчанию) — включить проверку подписи GPG для всех пакетов во всех репозиториях, включая локальную установку пакетов. При включенном gpgcheck проверяются все подписи пакетов.

Если этот параметр указан в разделе [main] файла /etc/yum.conf, это установит обязательность проверки GPG для всех репозиториев. Тем не менее, вместо этого можно установить gpgcheck=<значение> для индивидуальных репозиториев; то есть таким образом можно включить проверку GPG для одного репозиториях, а для другого — отключить ее. Установка gpgcheck=<значение> для отдельного репозитория в соответствующем файле .repo имеет приоритет над значениями по умолчанию, если они присутствуют в файле /etc/yum.conf.

group_command=<значение>

Используйте параметр group_command для определения того, как команды yum group install, yum group upgrade и yum group remove обрабатывают группу пакетов. Замените <значение> на одно из следующих:

- simple — установка всех пакетов, входящих в группу. Обновление только ранее установленных пакетов, но не устанавливать пакеты, добавленные в группу за это время;
- compat — аналогично simple, но помимо того yum upgrade также поставит пакеты, добавленные в группу с момента предыдущего обновления;
- objects (по умолчанию) — с этим параметром *yum* отслеживает ранее установленные группы и делает различия между пакетами, устанавливаемые в составе группы и пакетами, установленными отдельно.

group_package_types=<тип_пакета> [<еще_типы_пакетов>]

Здесь можно указать, какие типы пакетов (по выбору, по умолчанию или обязательные) будут устанавливаться при вызове команды yum group install. Типы по умолчанию и обязательные выбираются по умолчанию.

history_record=<значение>

Этот параметр указывает, что *yum* должен записывать историю транзакций. Замените <значение> на одно из следующих:

- 0 — *yum* не должен записывать историю транзакций;
- 1 (по умолчанию) — *yum* должен записывать историю транзакций.

Эта операция требует некоторого места на диске и некоторого дополнительного времени во время проведения транзакций, но представляет большой объем информации о проведенных операциях, которую можно просмотреть с помощью команды yum history. Значение history_record=1 является значением по умолчанию. Подробности о команде yum history см. в подразделе 3.9. «Работа с историей транзакций yum».

Примечание. Для обнаружения изменений, внесенных в базу данных rpmdb вне предела действия *yum*, используются записи истории транзакций. В этом случае *yum* показывает предупреждение и выполняет автоматический поиск возможных проблем, возникших из-за изменений в rpmdb. При отключенном `history_record` *yum* не имеет возможности определить такие изменения, и автоматическая проверка не выполняется.

installonlypkgs=<список_пакетов,_разделенных_пробелами>

Здесь можно указать список пакетов, разделенных пробелами, которые *yum* может установить, но никогда не будет обновлять. Список пакетов, которые по умолчанию можно только установить, см. на странице руководства yum.conf(5).

Если в файл /etc/yum.conf добавлена директива `installonlypkgs`, необходимо убедиться, что в ней указаны все пакеты, предназначенные только для установки, включая любые из пакетов, перечисленных в разделе `installonlypkgs` руководства yum.conf(5). Пакеты ядер, в частности, всегда должны быть указаны в `installonlypkgs` (по умолчанию они указаны), а значение параметра `installonly_limit` всегда должно быть больше 2, чтобы запасное ядро всегда было доступно в случае сбоя при загрузке ядра по умолчанию.

installonly_limit=<значение>

Этот параметр указывает, сколько именно пакетов, перечисленных в директиве `installonlypkgs`, можно установить одновременно. Замените `<значение>` целым числом, представляющим максимальное число версий, которое может быть установлено одновременно для каждого отдельного пакета, указанного в `installonlypkgs`. Значение по умолчанию для директивы `installonlypkgs` включает в себя несколько разных пакетов ядер, поэтому при изменении значения `installonly_limit` учитывайте, что оно также влияет на число установленных версий любого отдельного пакета ядра. Значение по умолчанию, указанное в /etc/yum.conf, — `installonly_limit=3`, и крайне не рекомендуется уменьшать это значение, особенно ниже 2.

keepcache=<значение>

Параметр `keepcache` определяет, должен ли *yum* хранить кэш заголовков и пакетов после успешной установки. Возможные значения:

- 0 (по умолчанию) — не оставлять кэш заголовков и пакетов после успешной установки;
- 1 — оставлять кэш после успешной установки.

logfile=<имя_файла>

Чтобы указать место для сбора сообщений журнала, замените `<имя_файла>` абсолютным путем до файла, в который *yum* должен записывать сообщения своего журнала. По умолчанию это /var/log/yum.log.

max_connenctions=<число>

Здесь `<число>` означает максимальное количество одновременных подключений, по умолчанию — 5.

multilib_policy=<значение>

Параметр `multilib_policy` настраивает параметры установки, если пакет под-

держивает несколько версий архитектуры. Возможные значения:

- `best` — установить лучший вариант архитектуры для данной системы. Если, например, установить `multilib_policy=best` на системе AMD64, *yum* будет устанавливать 64-битные версии всех пакетов;
- `all` — всегда устанавливать все доступные варианты архитектур для каждого пакета. Если, например, в системе AMD64 указать `all` для `multilib_policy`, *yum* установит как версию i686, так и AMD64, если они обе доступны.

obsoletes=<значение>

Параметр `obsoletes` включает логику обработки устаревших компонентов во время обновления. Если в файле spec одного пакета указано, что он делает другой пакет устаревшим, этот другой пакет будет заменен первым при установке первого пакета. Параметры `obsoletes` объявляются, например, при переименовании пакета. Возможные значения:

- 0 — отключить логику обработки устаревших компонентов во время выполнения обновления;
- 1 (по умолчанию) — включить логику обработки устаревших компонентов во время выполнения обновления.

plugins=<значение>

Это глобальный переключатель для включения или отключения модулей *yum*. Возможные значения:

- 0 — отключить все модули *yum* глобально;
- 1 — включить все модули *yum* глобально.

Примечание. Глобальное отключение модулей не рекомендуется, т. к. некоторые из модулей *yum* предоставляют важные службы *yum*. Эта возможность предоставляется в качестве вспомогательной функции и обычно рекомендуется к применению только при диагностировании потенциальных проблем с *yum*.

reposdir=<каталог>

Здесь `<каталог>` — это абсолютный путь до каталога, содержащего файлы .repo. Все файлы .repo содержат информацию о репозиториях (аналогично разделам `[repository]` файла /etc/yum.conf). Всю информацию о репозиториях для создания главного списка репозиториев, использующихся при транзакциях, *yum* собирает из файлов .repo и раздела `[repository]` файла /etc/yum.conf. Если параметр `reposdir` не настроен, *yum* использует каталог по умолчанию /etc/yum.repos.d/.

retries=<значение>

Этот параметр указывает количество попыток получения файла перед тем, как *yum* вернет ошибку. Значение — 0 или выше. Установка значения 0 заставит *yum* повторять попытки бесконечно. Значение по умолчанию — 10.

Полный список доступных параметров `[main]` см. в разделе [main] OPTIONS man-страницы yum.conf(5).

3.10.2. Установка параметров [repository]

Раздел `[repository]`, где `repository` — это уникальный идентификатор репозитория, например, my_personal_repo (пробелы не допускаются), дает возможность настроить персональные репозитории. Для избежания конфликтов, пользовательские репозитории не должны содержать в названиях имена, используемые в названиях репозиториев ROSA. Ниже дается минимальный пример раздела `[repository]`:

```
[repository]
name=имя_репозитория
baseurl=url_репозитория
```

Каждый раздел `[repository]` должен содержать следующие директивы:

name=<имя_репозитория>

Здесь <имя_репозитория> — это строка в удобочитаемом виде, описывающая репозиторий.

baseurl=<url_репозитория>

Замените `<url_репозитория>` URL-адресом каталога, в котором хранится каталог repodata.

- если репозиторий доступен по HTTP, используйте http://path/to/repo;
- если репозиторий доступен по FTP, используйте ftp://path/to/repo;
- если это локальный репозиторий, используйте file:///path/to/local/repo;
- если для какого-то сетевого репозитория требуется базовая аутентификация HTTP, можно указать имя пользователя и пароль перед адресом URL в виде «имя_пользователя:
  пароль@ссылка». Если, например, репозиторий по адресу http://www.test.dom/repo/ требует имя пользователя «user» и пароль «password», baseurl можно указать в виде http://user:password@www.test.dom/repo/.

  Обычно этот URL является ссылкой HTTP, например:

  baseurl=http://path/to/repo/releases/$releasever/server/$basearch/os/

Обратите внимание, что в URL *yum* всегда разворачивает переменные $releasever, $arch и $basearch.

Другие удобные директивы `[repository]`:

enabled=<значение>

Это простой способ указать yum, игнорировать или использовать конкретный репозиторий. `<значение>` может быть одним из:

- `0` — не включать этот репозиторий в источники пакетов при выполнении обновлений и установок. Это простой способ быстро подключить или отключить репозитории, что бывает удобно, когда требуется одинарный пакет из такого репозитория, который не нужен при обновлениях или других установках пакетов;
- `1` — включить этот репозиторий в источники пакетов.

Включить или отключить репозитории также можно, передав yum параметр `--enablerepo=repo_name` или `--disablerepo=repo_name`, а также в окне «Добавить/Удалить» программы в окне утилиты PackageKit.

async=<значение>

Управляет параллельной загрузкой пакетов из репозиториев. Здесь `<значение>` может быть одним из:

- `auto` (по умолчанию) — параллельная загрузка используется при возможности, т. е. yum автоматически отключает эту возможность для репозиториев, созданных модулями, во избежание сбоев;
- `on` — для репозитория включена параллельная загрузка;
- `off` — параллельная загрузка для репозитория отключена.

Для раздела `[repository]` существует еще множество параметров, некоторые из низ аналогичны по форме и действию некоторым параметрам для `[main]`. Полный их список см. в разделе [repository] OPTIONS man-страницы yum.conf(5).

3.10.3. Просмотр текущей конфигурации yum

Чтобы просмотреть текущее значение глобальных параметров *yum* (то есть параметров, указанных в разделе `[main]` файла /etc/yum.conf), выполните следующую команду без параметров:

```
$ yum-config-manager
```

Чтобы просмотреть список различных разделов с параметрами, используйте команду в следующем виде:

```
$ yum-config-manager <раздел> ...
```

Также можно использовать шаблон выражения для показа параметров во всех совпадающих разделах:

```
$ yum-config-manager <шаблон_выражения> ...
```

Например, чтобы получить список параметров и их соответствующих значений для раздела main, выполните:

```
$ yum-config-manager main \*
```

3.10.4. Добавление репозитория yum

Для определения нового репозитория можно добавить раздел `[repository]` либо в файл /etc/yum.conf, либо в файл .repo в каталоге /etc/yum.repos.d/. *Yum* читает все файлы с расширением .repo в этом каталоге, и рекомендуется настраивать репозитории здесь, а не в /etc/yum.conf.

Репозитории *yum* обычно предоставляют свой собственный файл .repo. Чтобы добавить такой репозиторий в свою систему и подключить его, выполните следующую команду с привилегиями суперпользователя root:

```
yum-config-manager --add-repo <url_репозитория>
```

Здесь `<url_репозитория>` — это ссылка на файл .repo.

3.10.4.1. Добавление example.repo

Чтобы добавить репозиторий, расположенный по адресу http://www.test.dom/example.repo, выполните следующую команду с привилегиями суперпользователя root:

```
# yum-config-manager --add-repo http://www.test.dom/example.repo
Loaded plugins: langpacks, product-id, subscription-manager
adding repo from: http://www.test.dom/example.repo
grabbing file http://www.test.dom/example.repo to
/etc/yum.repos.d/example.repo
example.repo                                             |  413 B
00:00
repo saved to /etc/yum.repos.d/example.repo
```

3.10.5. Включение репозитория yum

Чтобы включить конкретный репозиторий или репозитории, выполните следующую команду с привилегиями суперпользователя root:

```
# yum-config-manager --enable <репозиторий>
```

Здесь `<репозиторий>` — это уникальный идентификатор репозитория (используйте команду `yum repolist all` для получения списка доступных идентификаторов репозиториев). Также для включения всех совпадающих репозиториев можно использовать шаблон выражения:

```
# yum-config-manager --enable <шаблон_выражения>
```

3.10.5.1. Включение репозиториев, указанных в пользовательских разделах файла /etc/yum.conf

Чтобы включить репозитории из разделов `[example]`, `[example-debuginfo]` и `[example-source]`, выполните:

```
# yum-config-manager --enable example\*
```

3.10.5.2. Включение всех репозиториев

Чтобы включить все репозитории — как указанные в файле /etc/yum.conf, так и указанные в файле /etc/yum.repos.d/, выполните:

```
# yum-config-manager --enable \*
```

При успешном результате команда `yum-config-manager --enable` покажет текущую конфигурацию репозиториев.

3.10.6. Отключение репозиториев yum

Чтобы отключить репозиторий *yum*, выполните следующую команду:

```
yum-config-manager --disable <репозиторий>
```

Здесь `<репозиторий>` — уникальный идентификатор репозитория (используйте команду `yum repolist all` для получения списка доступных идентификаторов репозиториев). Как и для команды `yum-config-manager --enable`, для отключения всех совпадающих репозиториев можно использовать шаблон выражения:

```
yum-config-manager --disable <шаблон_выражения>
```

3.10.6.1. Отключение всех репозиториев

Чтобы отключить все репозитории — как указанные в файле /etc/yum.conf, так и указанные в файле /etc/yum.repos.d/, выполните:

```
# yum-config-manager --disable \*
```

При успешном результате команда `yum-config-manager --enable` покажет текущую конфигурацию репозиториев.

3.10.7. Создание репозитория yum

Чтобы создать репозиторий *yum*, выполните следующие действия:

1) Установите пакет createrepo:
   ```
   # yum install createrepo
   ```
2) Скопируйте все пакеты, которые должны быть в репозитории, в один каталог, например /mnt/local_repo/.
3) Перейдите в этот каталог и выполните следующую команду:
   ```
   createrepo --database /mnt/local_repo
   ```

Это действие создаст необходимые метаданные для созданного репозитория *yum*, а также базу данных sqlite для ускорения операций *yum*.

Локальные источники информации о *yum*:

- yum(8) — man-страница для консольной утилиты *yum*, предоставляющая полный список поддерживаемых параметров и команд;
- yumdb(8) — man-страница для консольной утилиты *yumdb*, описывающая использование этой утилиты для составления запросов в базу данных *yum*, и, при необходимости, изменение базы данных yum;
- yum.conf(5) — man-страница yum.conf, описывающая доступные параметры *yum*;
- yum-utils(1) — man-страница yum-utils, содержащая список и короткое описание дополнительных утилит для работы с параметрами *yum*, управления репозиториями и работы с базой данных *yum*.

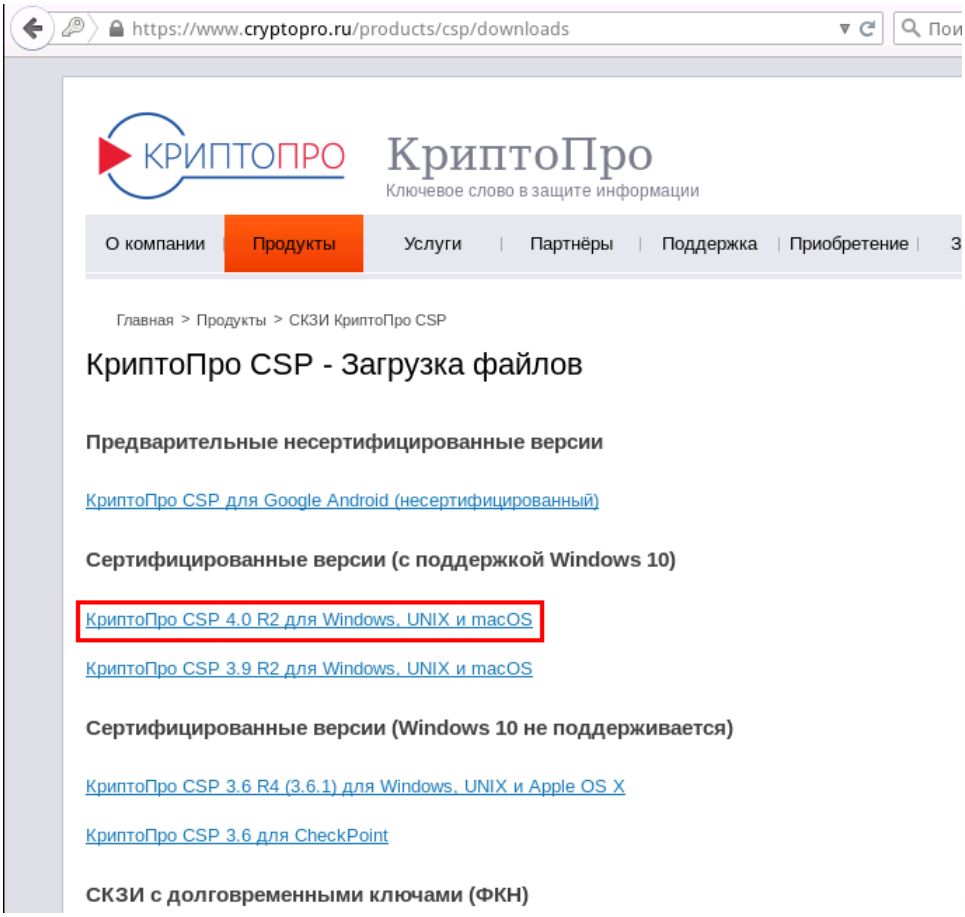
4. ИСПОЛЬЗОВАНИЕ ПОПУЛЯРНЫХ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ И РАБОТЫ С ЭЦП

4.1. Установка СКЗИ КриптоПро

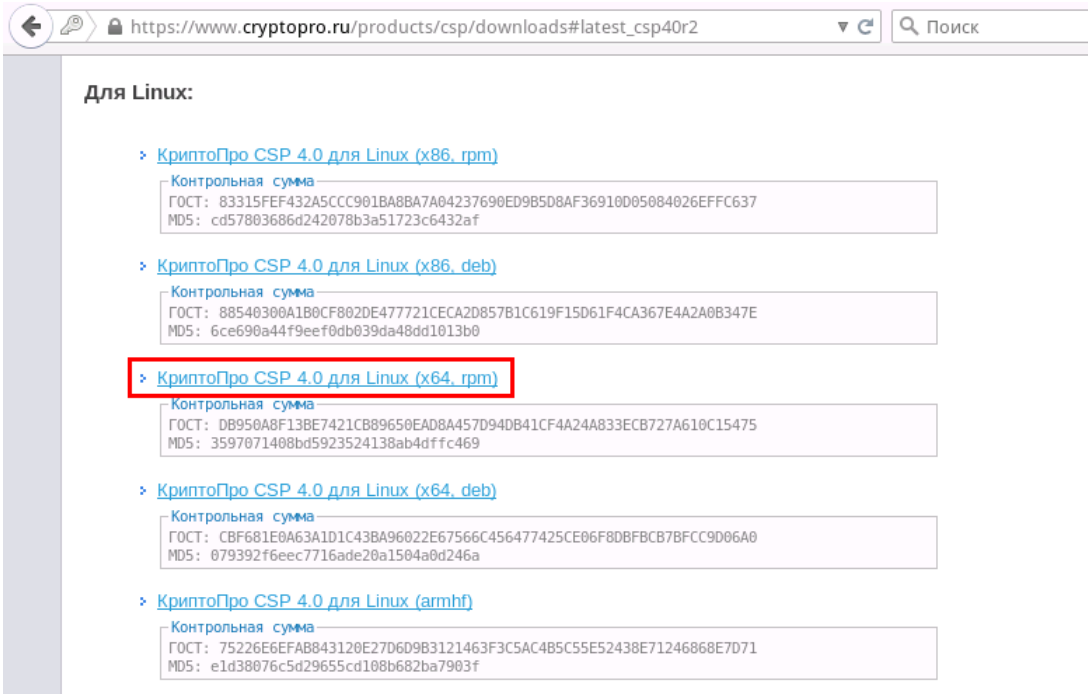
4.1.1. Получение установочных пакетов

Перед установкой СКЗИ КриптоПро CSP 4.0 следует зарегистрироваться на сайте https://www.cryptopro.ru/.

Далее со страницы загрузки https://www.cryptopro.ru/products/csp/downloads нужно скачать версию 4.0 последней ревизии для linux в формате rpm.



Страница КриптоПро для скачивания дистрибутива



Ссылка на rpm-пакет КриптоПро

4.1.2. Установка компонентов КриптоПро и связанных с ними пакетов

Примечание. Все команды и действия выполняются в терминале с правами обычного пользователя, которым будет в дальнейшем использован скачанный набор пакетов.

Для начала следует смонтировать устройство с образом системы (DVD либо USB-накопитель) в каталог /mnt/. Команда выполняется в терминале администратором системы:

```
sudo mount /dev/sr0 /mnt
```

Далее нужно распаковать архивы, скачивание которых описано в п. 4.1.1. «Получение установочных пакетов»:

```
cd ~/Загрузки
tar xvf linux-amd64.tgz
tar xvf cades_linux_amd64.tar.gz
```

Установите пакеты КриптоПро и пакеты, от которых они зависят:

```
cd linux-amd64
su -c 'yum -y install lsb pcsc-lite pangox-compat && ./install.sh
&& yum -y install cprocsp-rdr-pcsc-64-4.0.0-4.x86_64.rpm lsb-
cprocsp-pkcs11-64-4.0.0-4.x86_64.rpm cprocsp-rdr-gui-gtk-64-
4.0.0-4.x86_64.rpm'
```

Также следует установить пакеты с модулями поддержки и драйверами устройств, которые будут использованы вместе с КриптоПро. Названия пакетов с драйверами начинаются на ifd-. Названия пакетов с модулями поддержки начинаются на cprocsp-rdr-. Почти для каждого устройства нужен модуль поддержки, но не для всех необходимо устанавливать драйвера. Если в архиве есть модуль поддержки для вашего устройства, но нет драйвера, наиболее вероятно, что драйвер для него устанавливать не нужно.

Для более подробной информации по пакетам можно обратиться к документу «Руководство администратора безопасности Linux» ЖТЯИ.00087-01 93 03 из архива https://www.cryptopro.ru/sites/ default/files/docs/csp40/doc.zip.

Пример:

Для использования устройства Рутокен S необходимо установить как модуль поддержки, так и драйвер. Поэтому команда для установки имеет следующий вид:

```
su -c 'yum -y install ifd-rutokens-1.0.1-1.x86_64.rpm cprocsp-rdr-rutoken-64-4.0.0-4.x86_64.rpm'
```

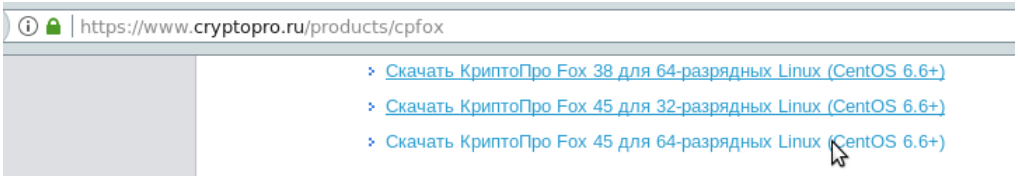
На этом установка КриптоПро CSP будет завершена.

После проделанных операций следует перезагрузить ОС.

4.2. Работа в браузере с поддержкой SSL/TLS ГОСТ

Для работы с КриптоПро Fox нужны установленные ранее пакеты из состава КриптоПро CSP и сам архив с КриптоПро Fox. Ниже описаны действия, подходящие для случая, когда выполнена установка КриптоПро CSP из предыдущего пункта. Для работы с КриптоПро Fox следует выключить Firefox.

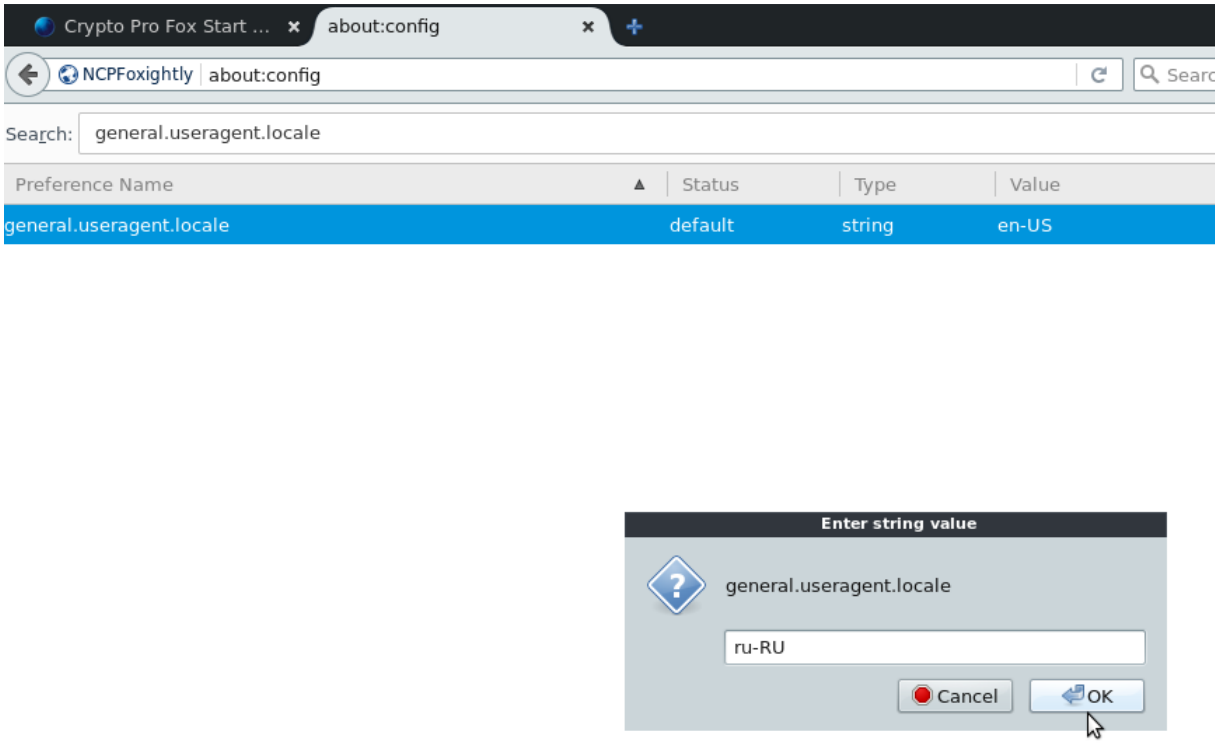
1) Перейдите на страницу продукта https://www.cryptopro.ru/products/cpfox и скачайте КриптоПро Fox 45.
2) Закройте Firefox.



Ссылки на КриптоПро Fox

3) Распакуйте архив в подходящую папку и запустите cpfox внутри распакованного архива. Для удобства можно создать ссылку на cpfox и переместить ее на рабочий стол.

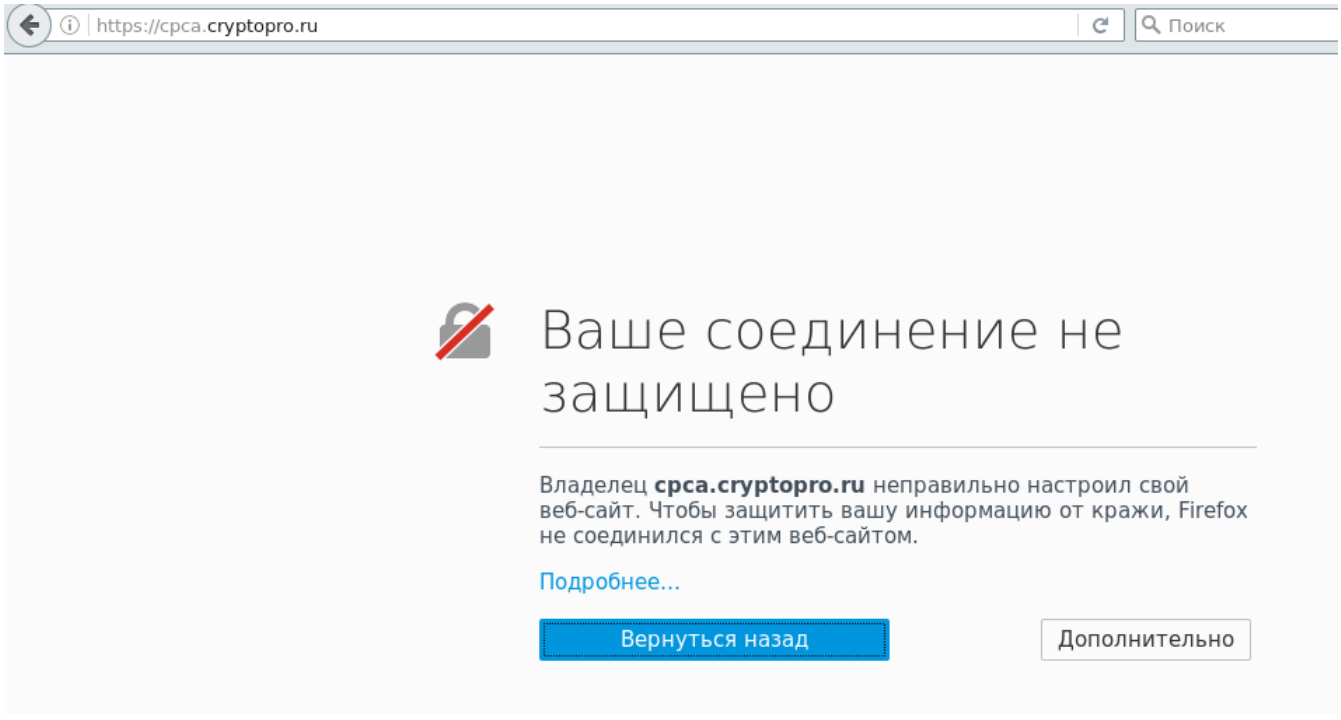
   Теперь можно работать в браузере КриптоПро Fox. Чтобы русифицировать его, перейдите на страницу https://addons.mozilla.org/ru/firefox/addon/russian-ru-language-pack/versions/ и нажмите на кнопку [Добавить в Firefox] напротив версии 45.0.
4) В адресной строке введите «about:config» и согласитесь со всплывающим сообщением.
5) Найдите при помощи окна «Search:» настройку с названием «general.useragent.locale». Нажмите на нее дважды и поменяйте значение на «ru-RU», затем нажмите [OK] и перезапустите КриптоПро Fox.



Русификация КриптоПро Fox

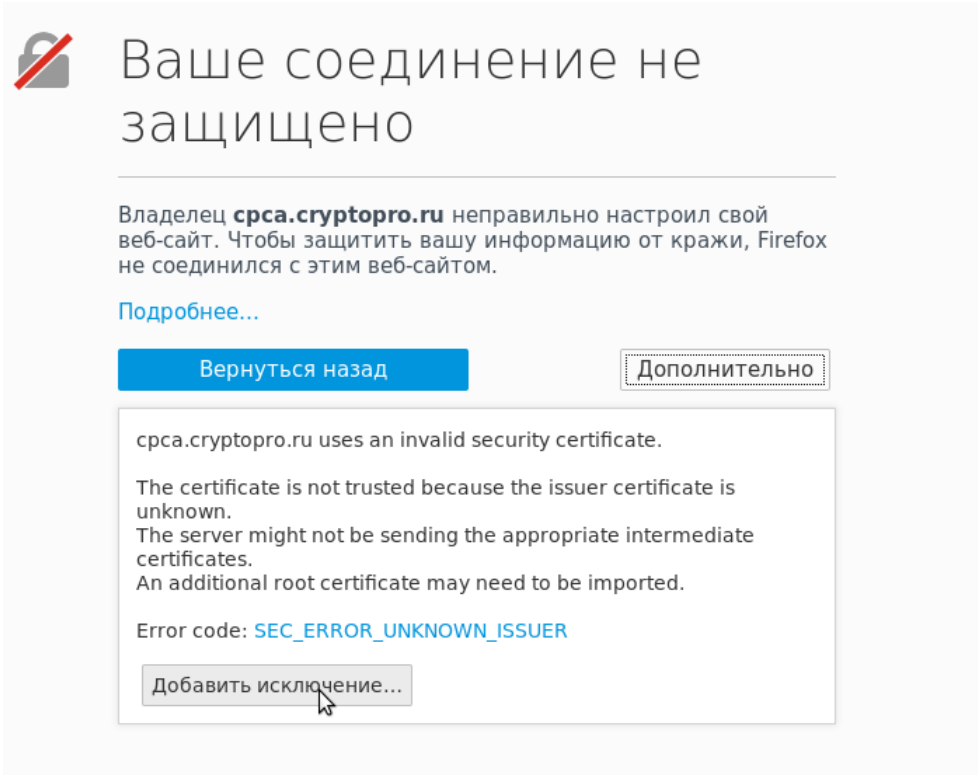
4.2.1. Проверка функционирования браузера

1) Для проверки функционирования перейдите на страницу https://cpca.cryptopro.ru/. В окне, отображающем информацию о странице, высветится информация о соединении.



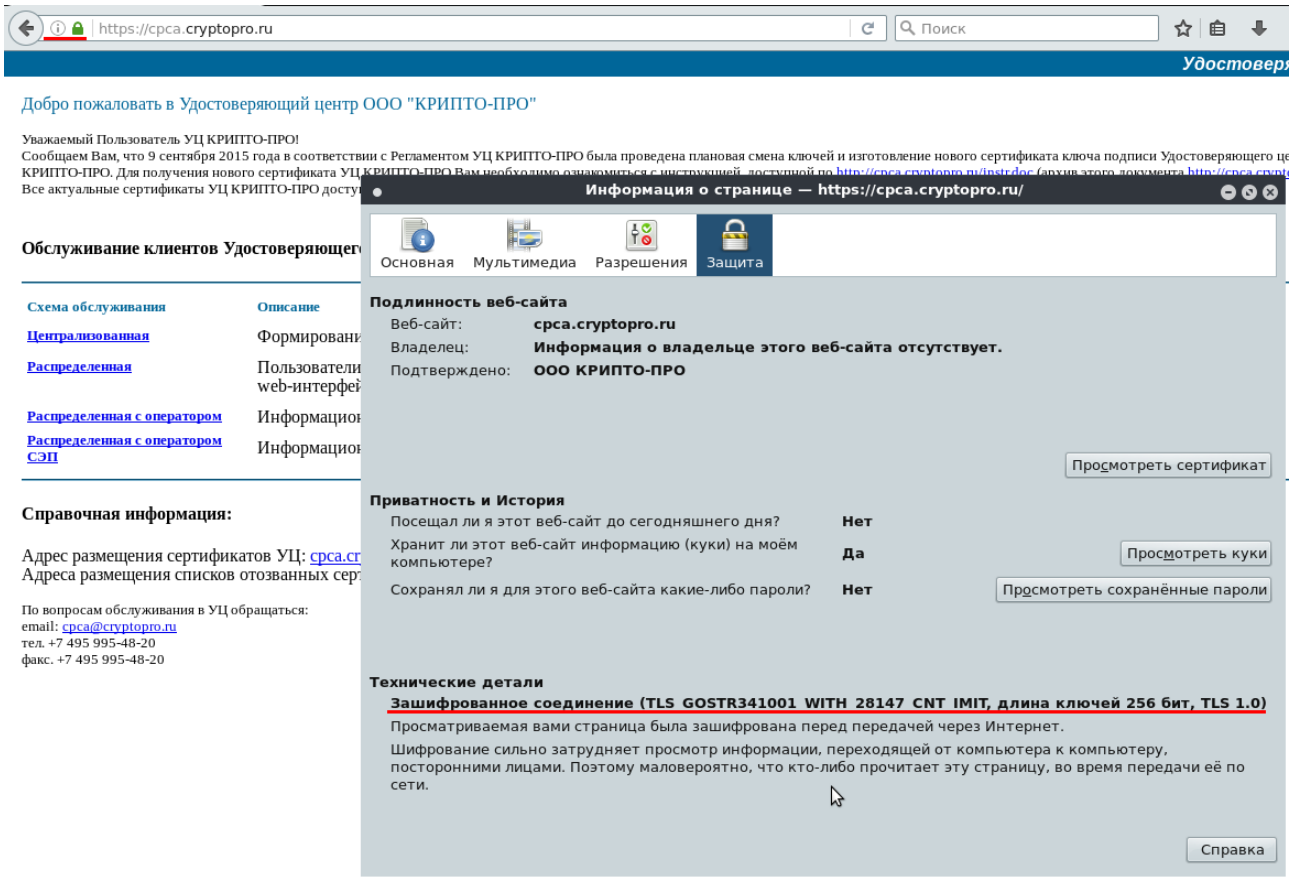
Сайт проверки работы КриптоПро Fox

2) Нажмите на кнопку [Дополнительно].



Меню «Дополнительно»

3) Нажмите на кнопку [Добавить исключение...].



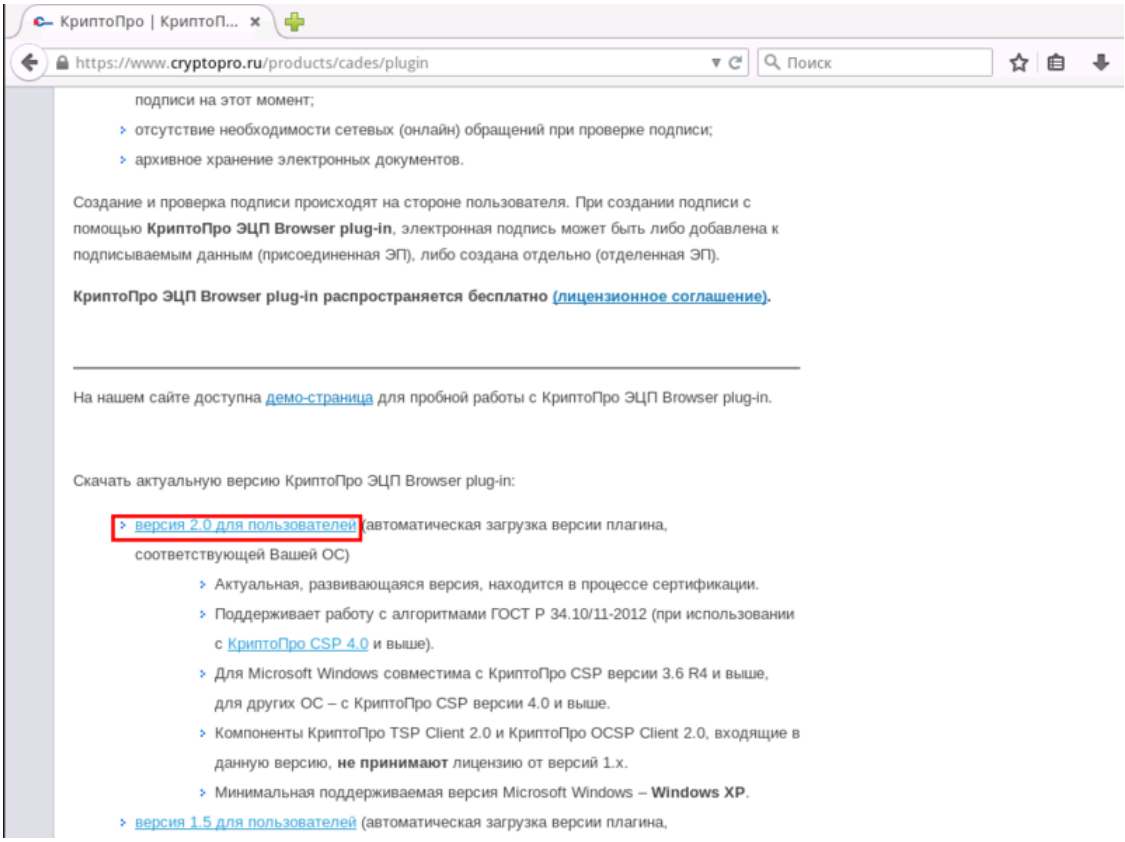
Проверка зашифрованного соединения

Если вы видите, что ваше соединение зашифровано, значит, КриптоПро Fox рабо-

тает корректно.

4.3. Настройка браузера для поддержки ЭЦП

Для скачивания КриптоПро ЭЦП Browser plug-in версии 2.0 перейдите на страницу https://cryptopro.ru/products/cades/plugin.



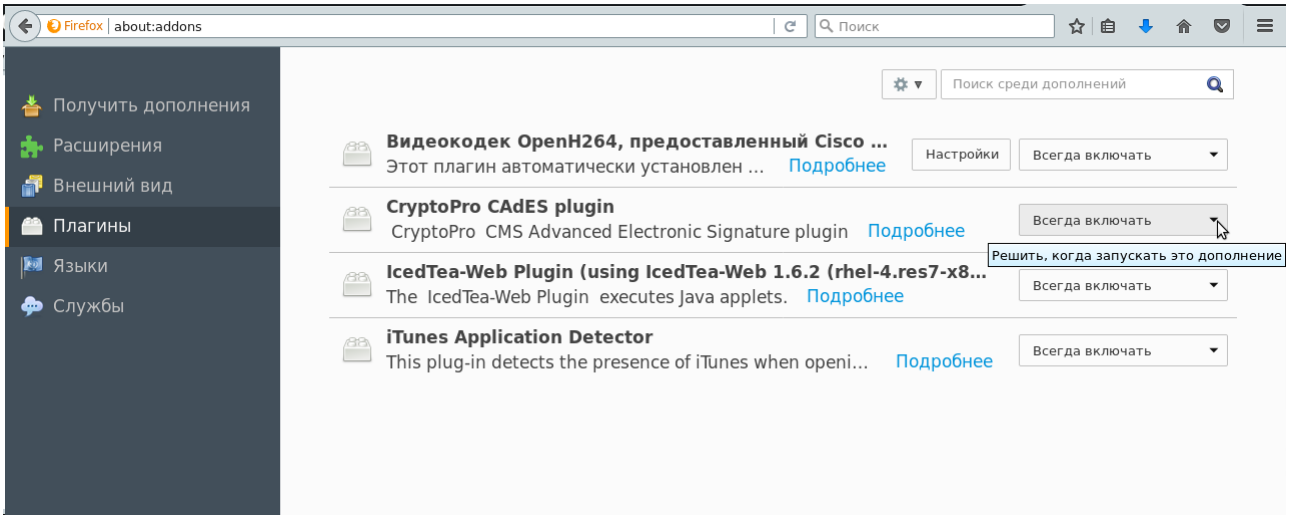
Загрузка плагина для браузера Firefox

Закройте браузер.

Для работы с КриптоПро ЭЦП Browser plug-in необходимо установить пакеты из архива, скачанного по данной ссылке: https://cryptopro.ru/products/cades/plugin. Для этого нужно ввести ряд команд от имени обычного пользователя, под которым вы загружали плагин:

```
cd ~/Загрузки
su -c 'yum -y install lsb-cprocsp-devel-4.0.0-4.noarch.rpm'
su -c 'yum -y install cprocsp-pki-2.0.0-amd64-cades.rpm cprocsp-pki-2.0.0-amd64-plugin.rpm'
```

После установки можно запустить Firefox и проверить наличие плагина. Следует также убедиться, что плагин включен (состояние «Всегда включать»).



Проверка включения плагина КриптоПро ЭЦП Browser plug-in

4.4. Создание тестового сертификата

Тестовые сертификаты КриптоПро можно получить на ресурсе, расположенном на сайте компании. Чтобы им воспользоваться, нужно добавить сайт КриптоПро в доверенные и установить корневой сертификат ресурса вместе со списком отозванных сертификатов.

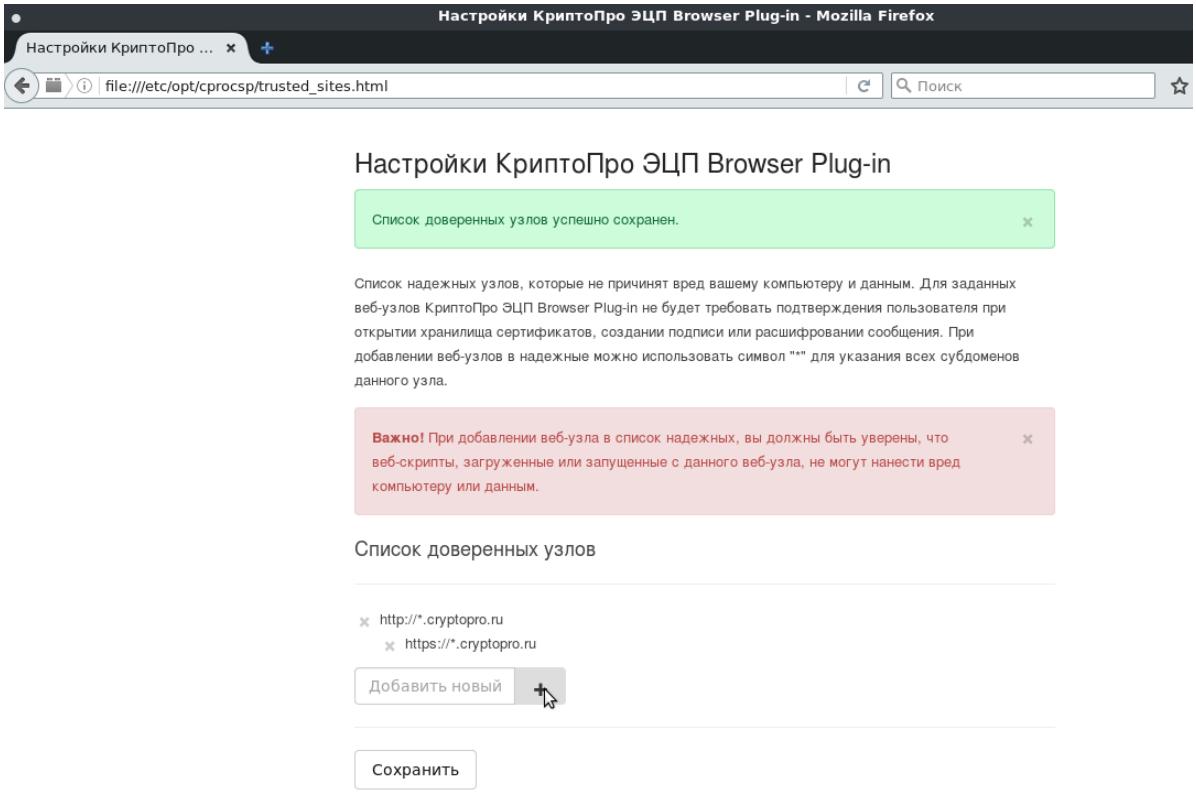
Далее:

1) Закройте браузер.
2) Выполните следующую команду:

   ```
   firefox /etc/opt/cprocsp/trusted_sites.html
   ```

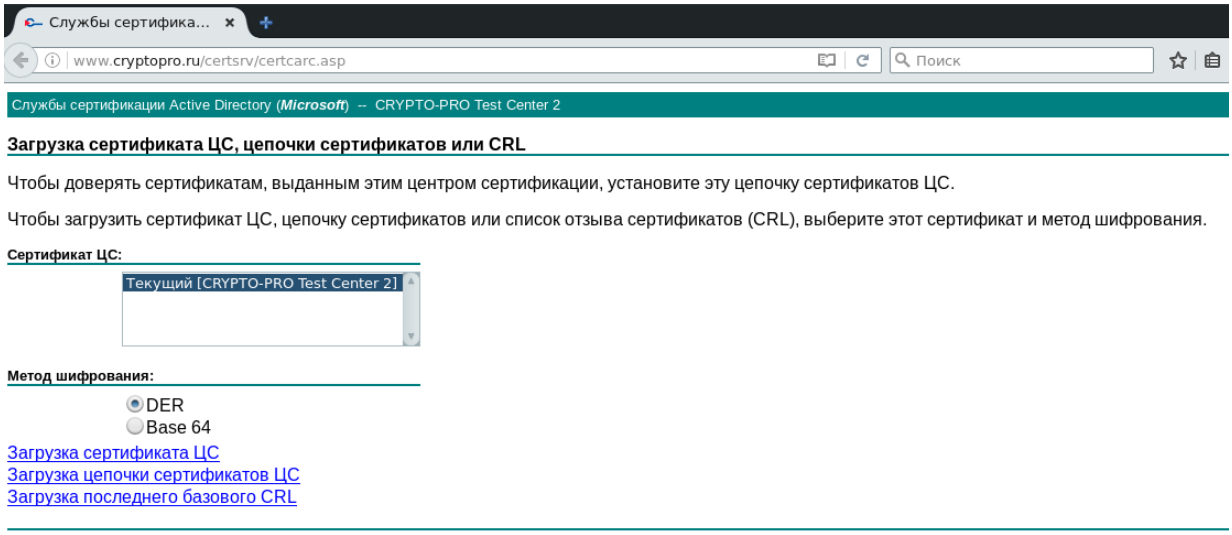
3) Добавьте в список доверенных два ресурса — http://*.cryptopro.ru и https://*cryptopro.ru.
4) Нажмите на кнопку [Сохранить].

Примечание. Следующие действия обязательно нужно выполнять от имени пользователя, который в дальнейшем будет использовать Browser Plug-in.



Настройки КриптоПро ЭЦП Browser Plug-in

5) Перейдите на страницу http://www.cryptopro.ru/certsrv/certcarc.asp и скачайте цепочку сертификатов и CRL («Загрузка цепочки сертификатов ЦС» и «Загрузка базового CRL»).



Загрузка цепочки сертификатов

6) Установите сертификаты:

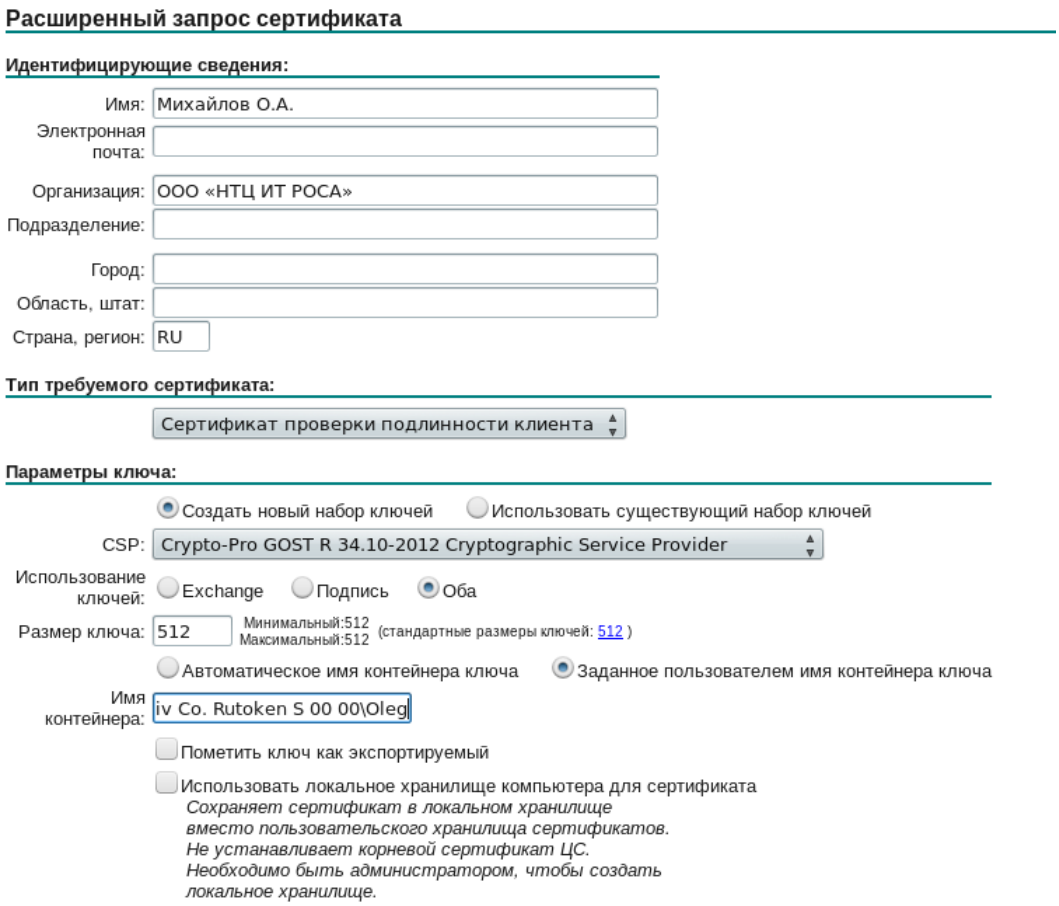
```
/opt/cprocsp/bin/amd64/certmgr -inst -cert -file
~/Загрузки/certnew.p7b -store uRoot &&
/opt/cprocsp/bin/amd64/certmgr -inst -crl -file
```

```
~/Загрузки/certcrl.crl
```

7) Перейдите на страницу http://www.cryptopro.ru/certsrv/certrqma.asp и укажите идентифицирующие сведения.
8) Параметры ключа, кроме имени контейнера, укажите сами. Необходимо выбрать пункт «Заданное пользователем имя контейнера». В появившемся окне нужно набрать полное имя контейнера, в котором будет храниться ключ с сертификатом. Это имя должно содержать название устройства. Название можно узнать следующей командой:

```
$ /opt/cprocsp/bin/amd64/list_pcsc
Aktiv Co. Rutoken S 00 00
```

9) Название контейнера должно иметь следующий вид: \\.\<название_устройства>\<имя_контейнера>.



Запрос на сертификат

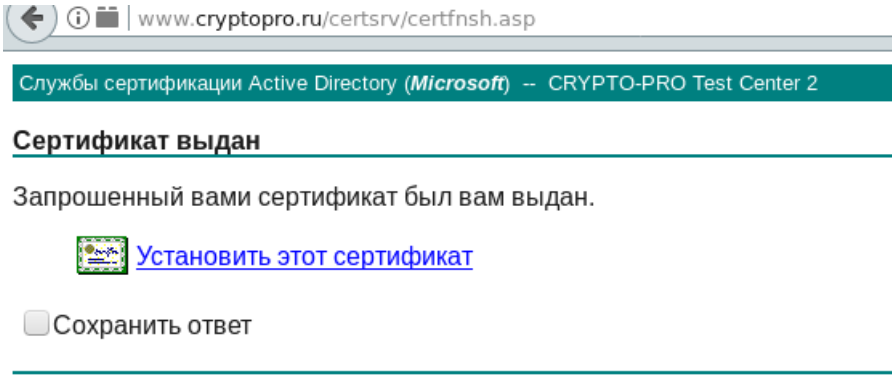
10) После заполнения всех полей нажмите на кнопку [Выдать >].
11) Появится окно датчика случайных чисел (ДСЧ). Двигайте мышью внутри окна или нажимайте кнопки для получения случайных данных.

Окно ДСЧ

12)После выработки случайной последовательности появится окно, запрашивающее пин-код к контейнеру. Введите пин-код, который в дальнейшем будет использоваться для доступа к контейнеру.

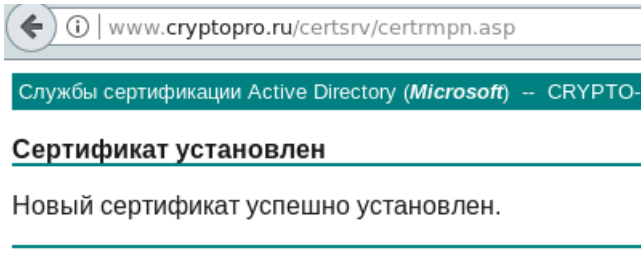
Ввод пин-кода к контейнеру

13)Нажмите на ссылку «Установить этот сертификат».



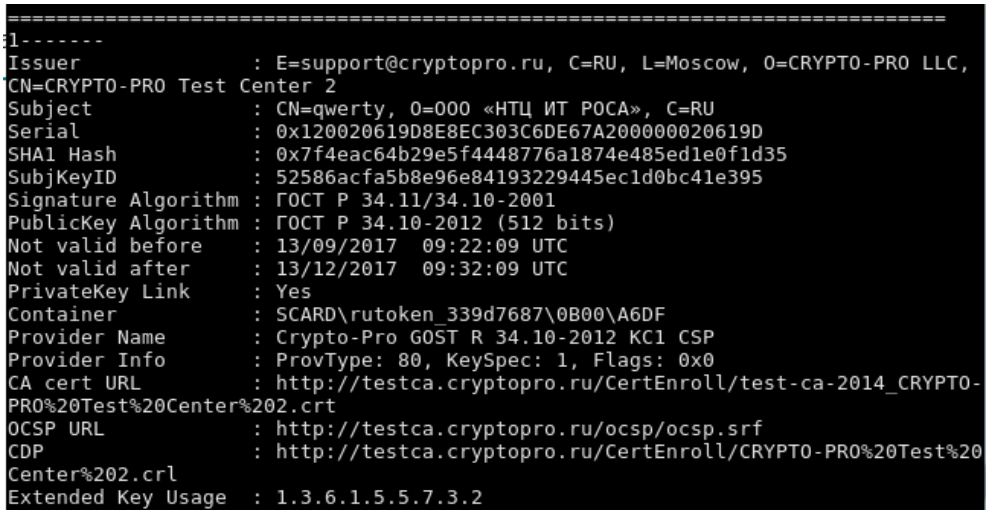
Установка сертификата

14) После этого снова появится окно, запрашивающее пин-код к контейнеру. Введите пин-код, придуманный ранее.



Успешная установка сертификата

После установки сертификата можно будет увидеть его при помощи команды `/opt/cprocsp/bin/amd64/certmgr -list`:



Информация о сертификате

Примечание. Если контейнер с указанным именем не создается, можно выбрать опции «Автоматическое имя контейнера ключа» и «Пометить ключ как экспортируемый». В таком случае появится возможность после генерации контейнера записать его на требуемое устройство самостоятельно, используя возможности КриптоПро CSP. Для этого можно воспользоваться следующей инструкцией:

http://wiki.rosalab.ru/ru/index.php/Создание_тестового_сертификата_КриптоПро.

В инструкции можно сразу перейти к пункту 2 пути 1.

4.5. Проверка работы КриптоПро ЭЦП Browser plug-in

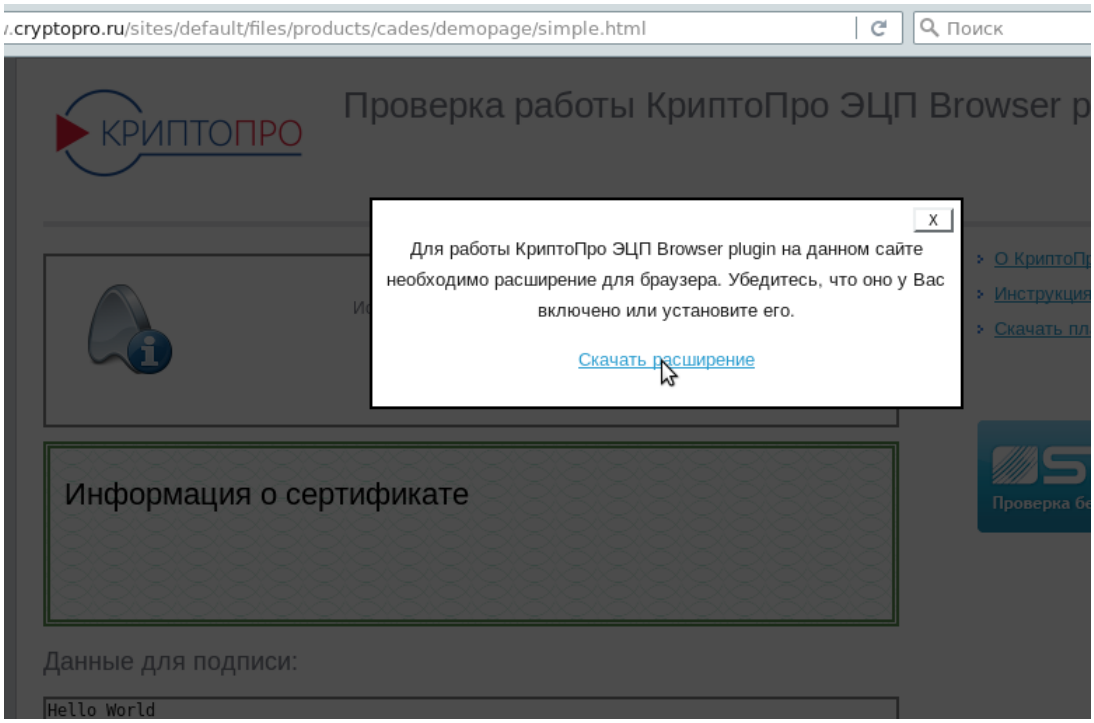
Для проверки работы плагина нужно существует специальный ресурс, расположенный на сайте КриптоПро. Чтобы им воспользоваться, нужно добавить сайт КриптоПро в доверенные и установить корневой сертификат ресурса вместе со списком отозванных сертификатов, как было описано выше.

1) В браузере перейдите на страницу

https://www.cryptopro.ru/sites/default/files/products/cades/ demopage/simple.html.

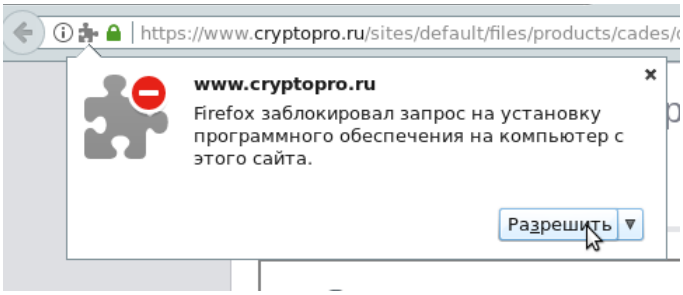
После недолгого ожидания сайт предложит установить расширение.

2) Нажмите на ссылку «Скачать расширение».

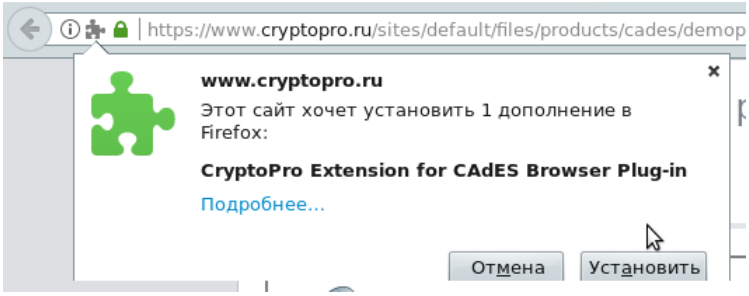


Загрузка расширения для браузера

3) Скачайте и установите расширение.

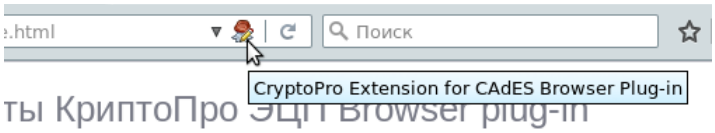


Запрос на установку расширения для браузера



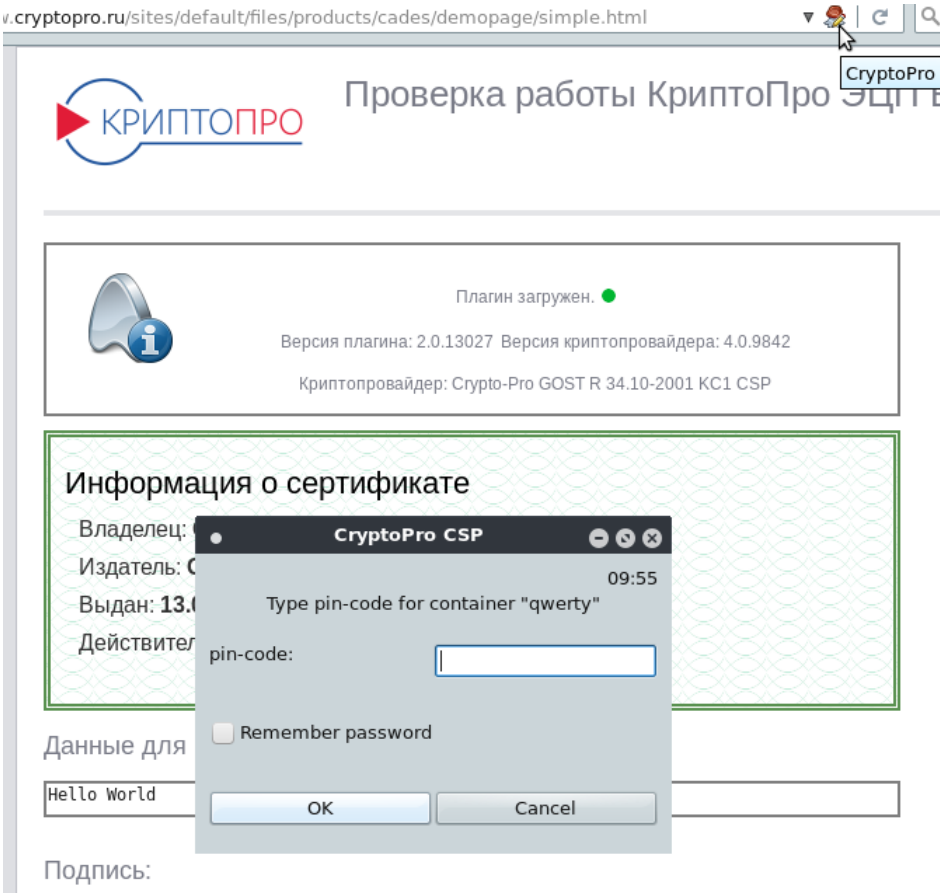
Установка расширения для браузера

После установки расширения должен появиться значок, представленный на рисунке ниже.



Значок расширения для браузера

4) Введите пин-код к контейнеру.



Ввод пин-кода к контейнеру

В нижней части страницы будет показана подпись сообщения «Hello World». Это инициирует успешную работу Browser Plug-in.

Криптопровайдер: **Crypto-Pro GOST R 34.10-2012 KC1 CSP**

Алгоритм ключа: **ГОСТ Р 34.10-2012**

Данные для подписи:

```
Hello World
```

Подпись сформирована успешно:

```
MIIIOgYJKoZIhvcNAQcCoIIIKzCCCCcCAQExDjAMBggqhQMHAQECAgUAMCUGCSqGSIb3DQEHAaAY
BBZIAGUAbABsAG8AIABXAG8AcgBsAGQAoIIFkTCCAkwwggH7oAMCAQICECtuM1H9brKtSCACA8tb
oUEwCAYGKoUDAgIDMH8xIzAhBgkqhkiG9w0BCQEWFHN1cHBvcnRAY3J5cHRvcHJvLnJ1MQswCQYD
VQQGEwJSVTEPMA0GA1UEBxMGTW9zY293MRcwFQYDVQQKEw5DUllQVE8tUFJPIExMQzEhMB8GA1UE
AxMYQ1JZUFRPLVBSTyBUZXN0IENlbnRlciAyMB4XDTE0MDgwNTEzNDQyNFoXDTE5MDgwNTEzNTQw
M1owfzEjMCEGCSqGSIb3DQEJARYUc3VwcG9ydEBjcnlwdG9wcm8ucnUxCzAJBgNVBAYTAlJVMQ8w
DQYDVQQHEwZNb3Njb3cxFzAVBgNVBAoTDkNSWVBUTy1QUk8gTExDMSEwHwYDVQQDExhDUllQVE8t
UFJPIFRlc3QgQ2VudGVyIDIwYzAcBgYqhQMCAhMwEgYHKoUDAgIjAQYHKoUDAgIeAQNDAARA4FIK
3EfcKb3XIF8MSHzv1I0ediXQvVIFjq3NystmDKbp5vpQJuSSIG5wgEq81v0ONxDH5oUBsu3WYo65
t83HnqNRME8wCwYDVR0PBAQDAgGGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBBYEFBUxfLCNGt5m
1xWcSVKXFyS5AXqDMBAGCSsGAQQBgjcVAQQDAgEAMAgGBiqFAwICAwNBANjKHEvpYSBlRyzVyOo4
iN7N7HHIRb+/nOXe/1VadySZ7eC5ItGr5/dP5tBve49aKuNO6WQNkFAywfHjtEkSV9UwggM9MIIC
7KADAgECAhMSACBhnY6OwwPG3meiAAAAIGGdMAgGBiqFAwICAzB/MSMwIQYJKoZIhvcNAQkBFhRz
dXBwb3J0QGNyeXB0b3Byby5ydTELMAkGA1UEBhMCUlUxDzANBgNVBAcTBk1vc2NvdzEXMBUGA1UE
ChMOQ1JZUFRPLVBSTyBMTEMxITAfBgNVBAMTGENSWVBUTy1QUk8gVGVzdCBDZW50ZXIgMjAeFw0x
NzA5MTMwOTIyMDlaFw0xNzEyMTMwOTMyMDlaMEgxDzANBgNVBAMMBnF3ZXJ0eTEoMCYGA1UECgwf
0J7QntCeIMKr0J3QotCmINCY0KIg0KDQntCh0JDCuzELMAkGA1UEBhMCUlUwZjAfBggqhQMHAQEB
ATATBgcqhQMCAiQABggqhQMHAQECAgNDAARAO4jDPZSBFDtqQif74msQTvJ3LnlQHqXoqSfHW+NW
yu29WyxocRiCx5rSzQu471bqAY1AfbgSVOtgEtAhZ9aWBaOCAXEwggFtMA8GA1UdDwEB/wQFAwMH
8AAwEwYDVR0lBAwwCgYIKwYBBQUHAwIwHQYDVR0OBBYEFFJYas+luOluhBkyKURewdC8QeOVMB8G
A1UdIwQYMBaAFBUxfLCNGt5m1xWcSVKXFyS5AXqDMFkGA1UdHwRSMFAwTqBMoEqGSGh0dHA6Ly90
ZXN0Y2EuY3J5cHRvcHJvLnJ1L0NlcnRFbnJvbGwvQ1JZUFRPLVBSTyUyMFRlc3QlMjBDZW50ZXIl
MjAyLmNybDCBqQYIKwYBBQUHAQEEgZwwgZkwYQYIKwYBBQUHMAKGVWh0dHA6Ly90ZXN0Y2EuY3J5
cHRvcHJvLnJ1L0NlcnRFbnJvbGwvdGVzdC1jYS0yMDE0X0NSWVBUTy1QUk8lMjBUZXN0JTIwQ2Vu
dGVyJTIwMi5jcnQwNAYIKwYBBQUHMAGGKGh0dHA6Ly90ZXN0Y2EuY3J5cHRvcHJvLnJ1L29jc3Av
b2NzcC5zcmYwCAYGKoUDAgIDA0EApj0trCUTFp0CCFHK6bR4bh9VJuEiglEUexxnBak9t3I+Ok98
jU3iLhK/14X2MmHudlcddZA4AOTYunEuAAW6mTGCAlQwggJQAgEBMIGWMH8xIzAhBgkqhkiG9w0B
CQEWFHN1cHBvcnRAY3J5cHRvcHJvLnJ1MQswCQYDVQQGEwJSVTEPMA0GA1UEBxMGTW9zY293MRcw
FQYDVQQKEw5DUllQVE8tUFJPIExMQzEhMB8GA1UEAxMYQ1JZUFRPLVBSTyBUZXN0IENlbnRlciAy
AhMSACBhnY6OwwPG3meiAAAAIGGdMAwGCCqFAwcBAQICBQCgggFSMBgGCSqGSIb3DQEJAzELBgkq
hkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTE3MDkxMzA5NDExMlowLwYJKoZIhvcNAQkEMSIEIGEv
wZlTzIRgtTxvZo5WtZcrewyk2EC/Z9ZUFCxJSvttMIHmBgsqhkiG9w0BCRACLzGB1jCB0zCB0DCB
zTAKBggqhQMHAQECAgQg01xf8yRhlXPwRPi0RJGlx2g6hhwXOwaDPUP7SwRpqv0wgZwwgYSkgYEw
fzEjMCEGCSqGSIb3DQEJARYUc3VwcG9ydEBjcnlwdG9wcm8ucnUxCzAJBgNVBAYTAlJVMQ8wDQYD
VQQHEwZNb3Njb3cxFzAVBgNVBAoTDkNSWVBUTy1QUk8gTExDMSEwHwYDVQQDExhDUllQVE8tUFJP
IFRlc3QgQ2VudGVyIDICExIAIGGdjo7DA8beZ6IAAAAgYZ0wDAYIKoUDBwEBAQEFAARAL/ykYB2z
rSiBQx8eLOiqWMOaRYMAu26j5Z57ejqtvsr3wMi08+gUMyGt5eSoLeyJDKSRblaGK7HBbN20467T
rg==
```

Успешная инициализация плагина для браузера

5. ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКИХ УТИЛИТ СЗИ

Примечание. При выборе на этапе установки рабочего окружения MATE будет установлен ряд утилит средств защиты информации (СЗИ). Если вы используете окружение GNOME, данные утилиты придется установить вручную.

Утилиты доступны по пути «Приложения → Утилиты СЗИ» (для окружения MATE) и «Приложение → Другие» (для окружения GNOME).

5.1. Использование утилиты ROSA Crypto Tool

Утилита ROSA Crypto Tool установлена в дистрибутиве по умолчанию. Она используется в качестве графической оболочки для утилит командной строки, входящих в состав КриптоПро. Утилита предназначена для работы с электронной подписью и шифрованием.

Для работы с ROSA Crypto Tool необходимо сначала подключить устройство, а затем запустить саму программу.

5.1.1. Введение

Утилита ROSA Crypto Tool работает с электронно-цифровыми подписями, хранящимися в контейнере формата .sig СКЗИ КриптоПро.

В программе предусмотрена реализация подписи и проверки подписи файлов в соответствии с ГОСТ Р 34.10-2012 и ГОСТ Р 34.10-2001 (см. Приложение А на стр. 59).

5.1.2. Описание элементов интерфейса

Интерфейс программы выглядит следующим образом:

Пользовательский интерфейс ROSA Crypto Tool

Далее будут описаны основные компоненты рабочего окна программы.

5.1.2.1. Панель инструментов

На панели инструментов располагаются пять кнопок. Первые четыре кнопки предназначены для переключения режимов работы с СКЗИ, а именно:

1) Подписать файл.

2) Проверить подпись.
3) Шифровать.
4) Расшифровать.

На следующем рисунке представлена панель инструментов.

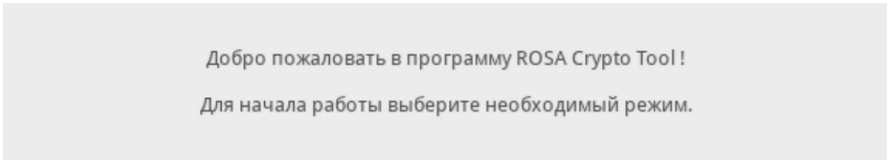
Панель инструментов ROSA Crypto Tool

Последняя кнопка называется [Параметры] и содержит в себе дополнительное подменю, не относящееся напрямую к работе с СКЗИ.

5.1.2.2. Рабочая область

Рабочая область программы располагается под панелью инструментов.

Если все компоненты, необходимые для полного функционирования программы, успешно установлены и функционируют, в рабочей области будет отображаться только приветствующий текст:



Рабочая область (приветствующий текст)

В противном случае под приветствующем текстом будет выведено соответствующее сообщение:

Рабочая область (приветствующий текст с заметкой)

При подключении или извлечении токена из компьютера под текстом приветствия будет выведено соответствующее сообщение.

После выбора кого-либо из режимов на панели инструментов в рабочей области по-

явится набор графических элементов для работы с СКЗИ.

На рисунке представлено поле, информирующее о статусе токенов.

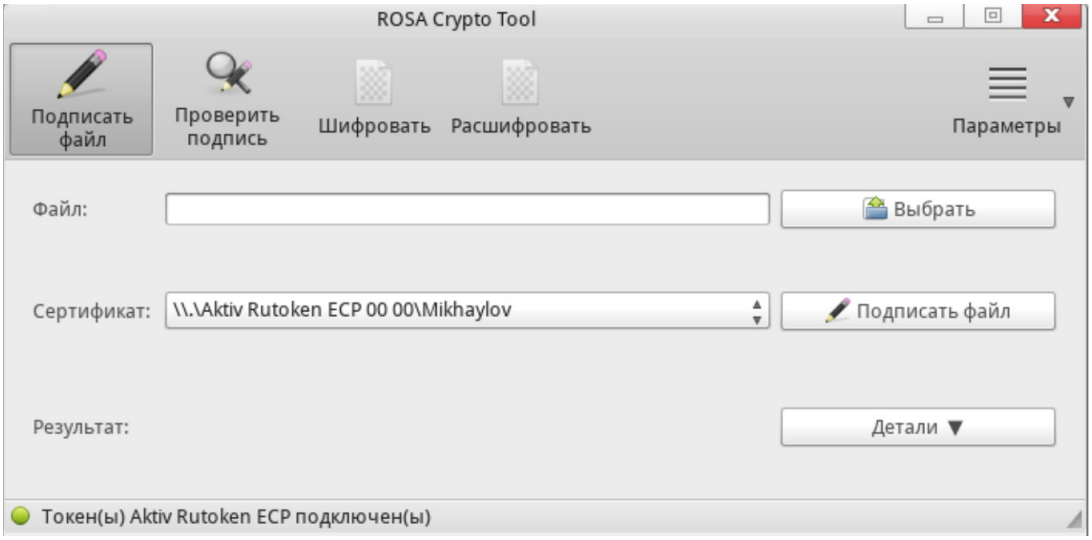
Статусное поле

Это поле доступно во всех режимах.

5.1.3. Подпись файла

Чтобы подписать файл, необходимо:

1) На панели инструментов выбрать режим «Подписать файл».
2) Указать файл с помощью кнопки [Выбрать].
3) Если в компьютере установлено несколько токенов, в поле «Сертификат» из выпадающего списка выбрать необходимый.
4) Нажать на кнопку [Подписать файл].



Режим «Подписать файл»

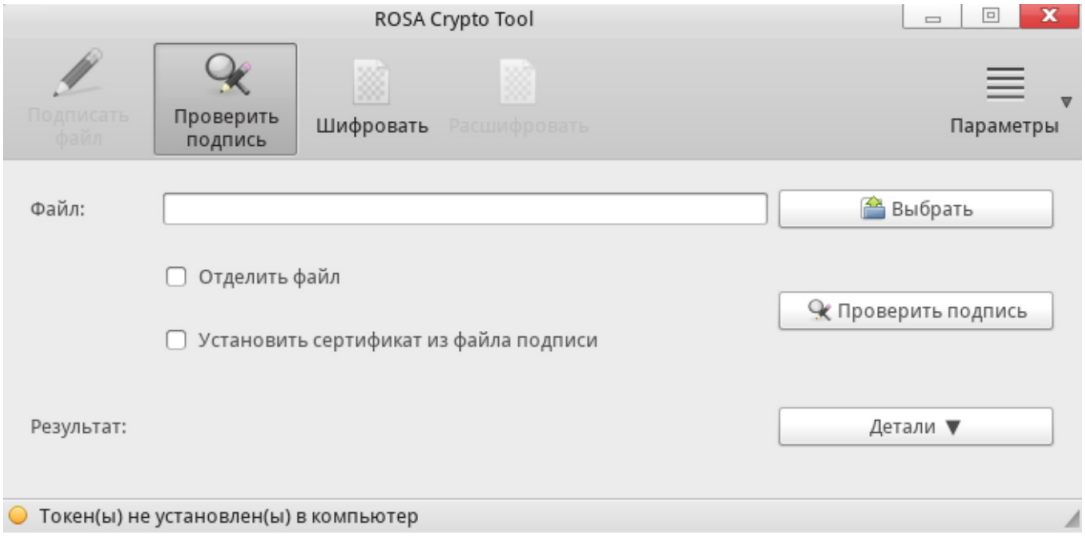
После успешного выполнения операции в поле «Результат» будет выведено соответствующее оповещение, и в папке выбранного файла появится подписанный файл с расширением .sig.

Кнопка [Детали] раскрывает поле «Результат» для отображения более полной информации, доступной для выделения и копирования.

5.1.4. Проверка подписи

Чтобы проверить подпись файла, необходимо:

1) На панели инструментов выбрать режим «Проверить подпись».
2) Указать файл с помощью кнопки [Выбрать].
3) Если дополнительно необходимо установить сертификат из файла подписи и/или отделить исходный файл от файла подписи, выставить галочки слева от имени соответствующей дополнительной опции.
4) Нажать на кнопку [Проверить подпись].



Режим «Проверить подпись»

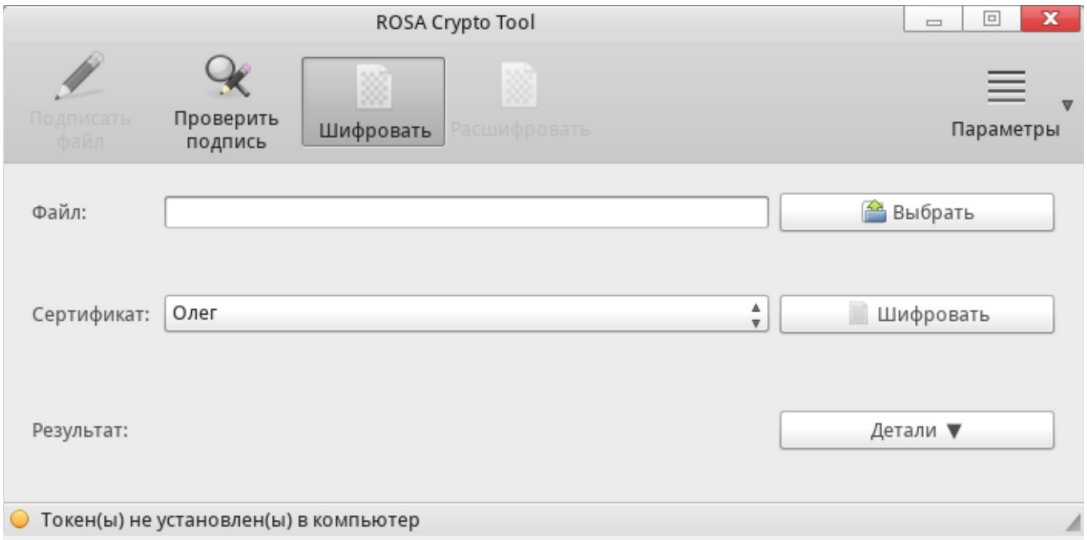
После выполнения операции в поле «Результат» будет выведено соответствующее оповещение.

Кнопка [Детали] раскрывает поле «Результат» для отображения более полной информации, доступной для выделения и копирования.

5.1.5. Шифрование файла

Чтобы выполнить шифрование файла, необходимо:

1) На панели инструментов выбрать режим «Шифровать».
2) Указать файл с помощью кнопки [Выбрать].
3) В поле «Сертификат» выбрать сертификат, с помощью которого необходимо зашифровать файл.
4) Нажать на кнопку [Шифровать].



Режим «Шифровать»

После успешного выполнения операции в поле «Результат» будет выведено соответствующее оповещение, и в каталоге выбранного файла появится файл подписи с

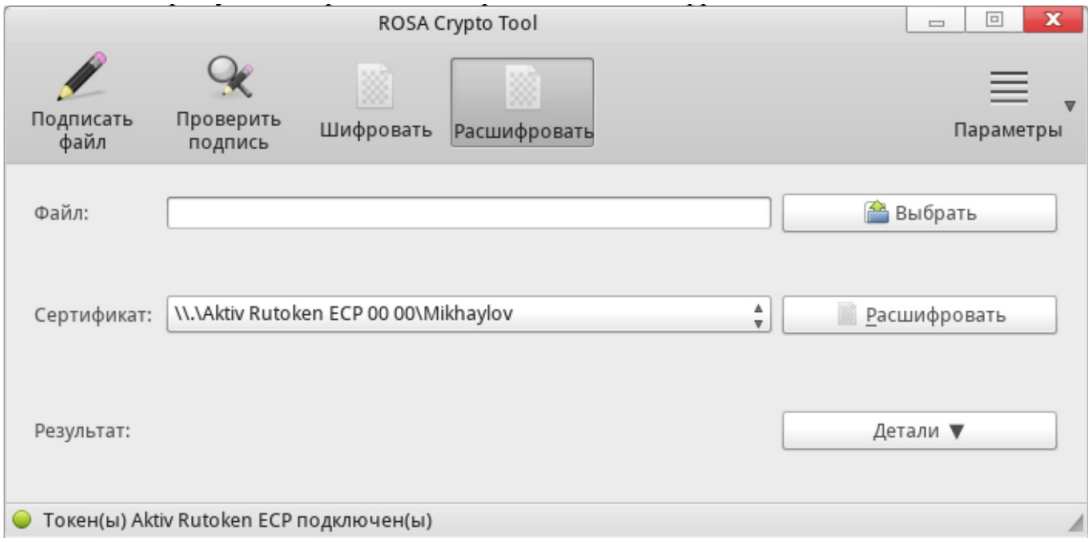
расширением .enc.

Кнопка [Детали] раскрывает поле «Результат» для отображения более полной информации, доступной для выделения и копирования.

5.1.6. Расшифрование файла

Чтобы выполнить расшифровывание файла, необходимо:

1) На панели инструментов выбрать режим «Расшифровать».
2) Указать файл с помощью кнопки [Выбрать].
3) Если в компьютере установлено несколько токенов, в поле «Сертификат» из выпадающего списка выбрать необходимый.
4) Нажать на кнопку [Расшифровать].



Режим «Расшифровать»

После выполнения операции в поле «Результат» будет выведено соответствующее оповещение.

Кнопка [Детали] раскрывает поле «Результат» для отображения более полной информации, доступной для выделения и копирования.

5.1.7. Параметры

Кнопка [Параметры] содержит в себе дополнительное подменю, включающее в себя такие опции, как:

- «Проверка компонентов программы» — проверяет наличие необходимых компонентов для успешной работы программы и соответствующее оповещение пользователя;
- «О программе ROSA Crypto Tool» — выводит краткую информацию о программе;
- «Справка» — открывает руководство пользователя;
- «Выход» — осуществляет выход из программы.

5.1.8. Приложение А

Порядок перехода к использованию национального стандарта ГОСТ Р 34.10-2012 в средствах электронной подписи для информации, не содержащей сведений, составляющих государственную тайну, в случаях, подлежащих регулированию со стороны ФСБ России в соответствии с действующей нормативной правовой базой (выписка из документа ФСБ России № 149/7/1/3-58 от 31.01.2014 «О порядке перехода к использованию новых стандартов ЭЦП и функции хеширования»).

Для средств ЭП, техническое задание на разработку которых утверждено после 31.12.2012, должна быть предусмотрена реализация функций средства в соответствии с ГОСТ Р 34.10-2012 хотя бы по одному из определяемых стандартом вариантов требований к параметрам (использование варианта, соответствующего длине секретного ключа порядка 256 бит, является предпочтительным, поскольку обеспечивает достаточный уровень криптографической стойкости и лучшие эксплуатационные характеристики, в том числе при совместной реализации со схемой ГОСТ Р 34.10-2001). После 31.12.2013 не осуществлять подтверждение соответствия средств ЭП Требованиям к средствам электронной подписи, утвержденным приказом ФСБ России от 27.12.2011 г. No 796, если в этих средствах не предусмотрена реализация функций средства в соответствии с ГОСТ Р 34.10-2012 хотя бы по одному из определяемых стандартом вариантов требований к параметрам. Исключение может быть сделано для средств ЭП, удовлетворяющих одновременно следующим условиям:

- техническое задание на разработку средства утверждено до 31.12.2012;
- в соответствии с техническим заданием разработка средства завершена после 31.12.2011;
- подтверждение соответствия средства указанным Требованиям ранее не осуществлялось.

Примечание. Использование схемы подписи ГОСТ Р 34.10-2001 для формирования подписи после 31.12.2018 не допускается.

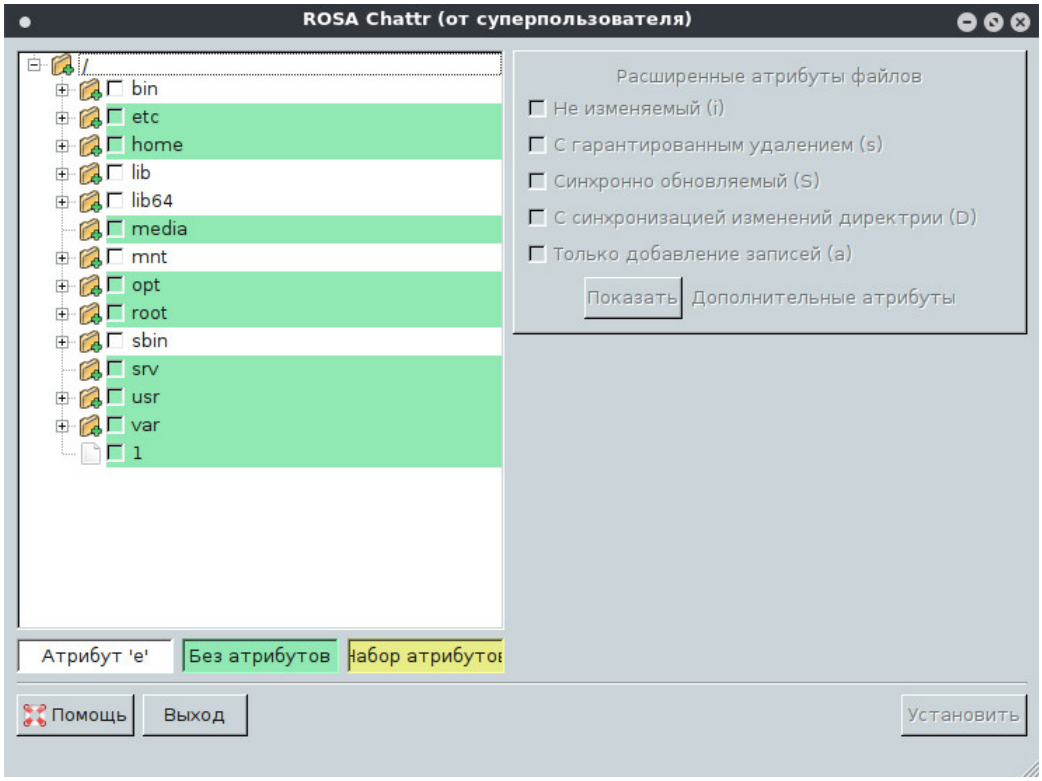
5.2. Использование утилиты ROSA Chattr

Для работы с программой требуются привилегии суперпользователя root.

5.2.1. Введение

Утилита ROSA Chattr предназначена для назначения и модификации дополнительных атрибутов файлов и каталогов. Чтобы запустить утилиту, выберите пункт меню «Приложения → Утилиты СЗИ ОС РОСА → ROSA Chattr» или выполните команду `rosa-chattr`.

Интерфейс программы выглядит следующим образом:

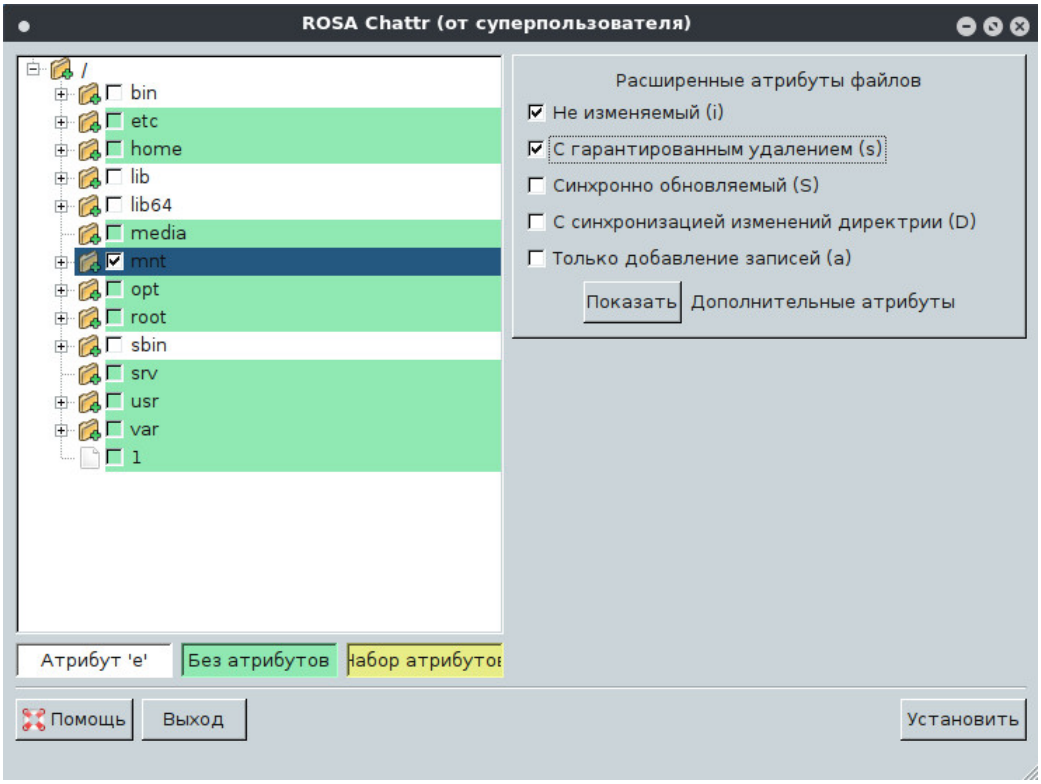


Интерфейс ROSA Chattr

В окне слева представлена корневая файловая система вашей ОС. Файлы и каталоги снабжены цветовой подсказкой в зависимости от наличия тех или иных атрибутов. Справа представлены атрибуты, которые можно установить. Слева внизу есть подсказка по цветовому обозначению.

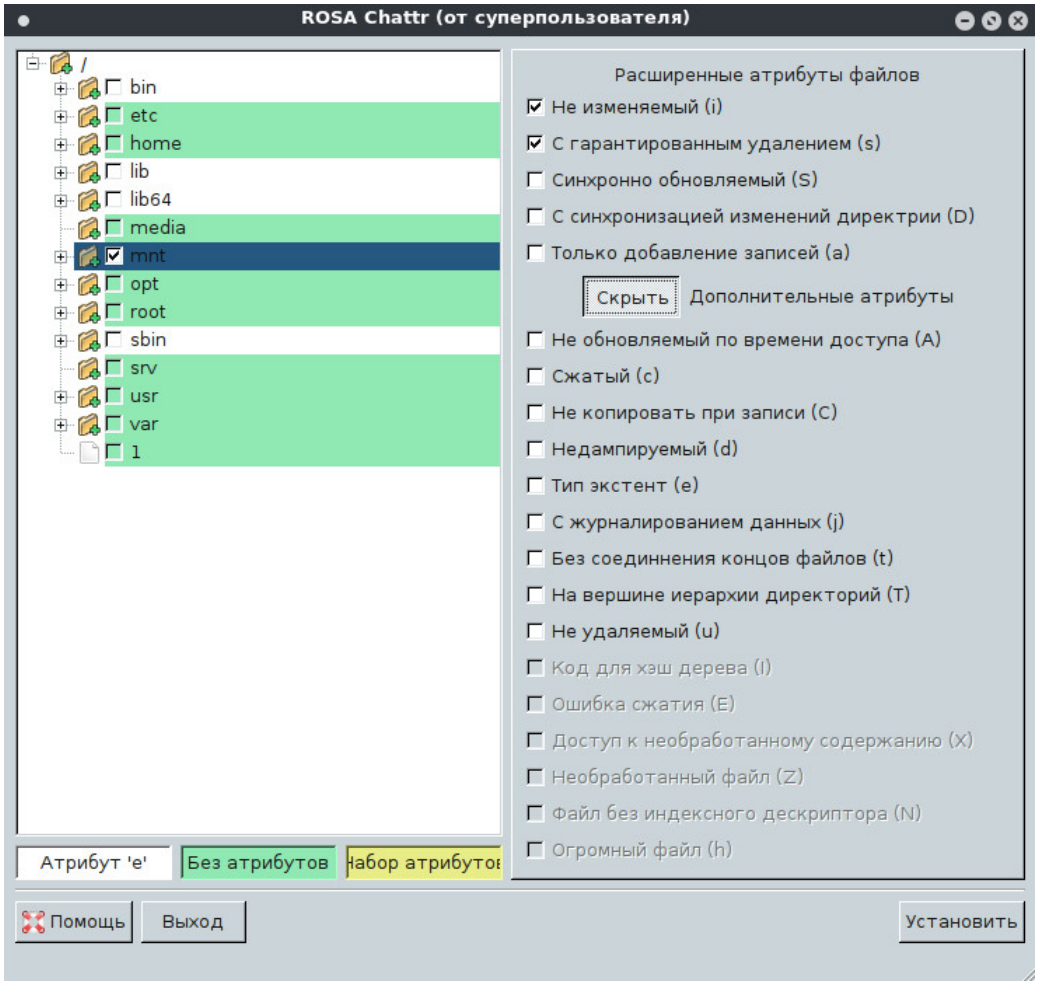
Пока не выбрано ни одного файла или каталога слева, выбор атрибутов будет невозможен.

Чтобы назначить расширенные права файлу или каталогу, выберите его, а затем установите для него необходимые атрибуты справа и нажмите на кнопку [Установить]. Атрибуты устанавливаются нерекурсивно.



Установка атрибутов

Если нажать на кнопку [Показать], будет доступен список дополнительных атрибутов:



Дополнительные атрибуты

5.2.2. Перечень атрибутов

1) Если для файла установлен атрибут «A», обновление (модификация) записи atime (времени доступа к файлу) не происходит. Это позволяет избежать дополнительных дисковых операций ввода/вывода.
2) Для файла с установленным атрибутом «a» разрешено лишь добавлять записи. Только суперпользователь или процесс, обладающий возможностью CAP_LINUX_IMMUTABLE, может установить или очистить этот атрибут.
3) Информация файла с установленным атрибутом «c» автоматически упаковывается (сжимается) на диске ядром операционной системы. Операция чтения информации из этого файла возвращает несжатые данные. Запись информации в такой файл сопровождается предварительной ее упаковкой и, наконец, последующим сохранением на диск.
4) При модификации файла с атрибутом «D» внесенные изменения синхронно записываются на диск; использование этого атрибута эквивалентно применению опции монтирования «dirsync» к подмножеству файлов.
5) Для файла с установленным атрибутом «d» не выполняется резервное копирование, когда запущена программа dump(8).
6) Атрибут «E» используется экспериментальными заплатками сжатия для определе-

ния того, что сжатый файл имеет ошибку сжатия. Это состояние может быть установлено или сброшено с помощью chattr(1), а отображено — с помощью lsattr(1).

7) Атрибут «I» используется кодом для хеш-деревьев (htree), чтобы указать, что каталог находится позади индексированных хешированных деревьев. Это состояние может быть установлено или сброшено с помощью chattr(1), а отображено — с помощью lsattr(1).
8) Файл с установленным атрибутом «i» становится немодифицируемым (недосягаемым): он не может быть удален или переименован, на этот файл не могут быть созданы никакие ссылки и никакие данные не могут быть записаны в него. Только суперпользователь или процесс, обладающий возможностью CAP_LINUX_IMMUTABLE, может установить или очистить этот атрибут.
9) Для файла с установленным атрибутом «j» все его данные, прежде чем быть записанными непосредственно в файл, сохраняются в журнал ext3. Правда, это происходит лишь в том случае, если файловая система была смонтирована с опциями «data=ordered» или «data=writeback». Когда файловая система смонтирована с опцией «data=journal», все данные файла уже журналируются, и этот атрибут не имеет никакого эффекта. Только суперпользователь или процесс, обладающий возможностью CAP_SYS_RESOURCE, может установить или очистить этот атрибут.
10) При удалении файла с установленным атрибутом «s» выполняется обнуление его блоков и запись их обратно на диск.
11) При модификации файла с атрибутом «S» внесенные изменения синхронно записываются на диск; использование этого атрибута эквивалентно применению опции монтирования «sync» к подмножеству расположенных файлов.
12) Каталог с установленным атрибутом «T» будет считаться расположенным на вершине иерархии каталогов с целью использования метода распределения блоков по Orlov (который применяется в системах с Linux 2.5.46 или выше).
13) Файл с установленным атрибутом «t» не будет иметь в завершающем блоке на диске дописанных («склеенных» с ним) фрагментов других файлов (для тех файловых систем, которые поддерживают «склеивание хвостов» файлов). Это необходимо для программ типа LILO, которые непосредственно обращаются к файловой системе и не понимают «склеивание хвостов». Здесь следует отметить, что файловые системы ext2 или ext3 по умолчанию не поддерживают для файлов «склеивание хвостов».
14) При удалении файла с атрибутом «u» его содержимое сохраняется (остается нетронутым) на диске. Это позволяет пользователю в дальнейшем восстановить такой файл.
15) Атрибут «X» используется экспериментальными заплатками сжатия для определения того, что к необработанному содержанию сжатого файла можно получить непосредственный доступ. Это состояние может быть установлено или сброшено с помощью chattr(1), а отображено — с помощью lsattr(1).
16) Атрибут «Z» используется экспериментальными средствами сжатия для определения того, что сжатый файл является необработанным. Это состояние может быть установлено или сброшено с помощью chattr(1), а отображено — с помощью

lsattr(1).

5.2.3. Ошибки и ограничения

Атрибуты «c», «s» и «u» пока не работают в файловых системах ext2 и ext3. Атрибут «j» полезен только для смонтированной файловой системы ext3.

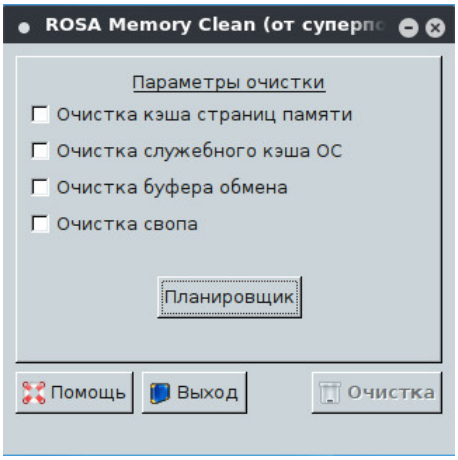
5.3. Использование утилиты ROSA Memory Clean

Для работы с программой требуются привилегии суперпользователя root.

5.3.1. Введение

Утилита ROSA Memory Clean предназначена для освобождения памяти ОС. Имеется возможность освобождения различных участков памяти по расписанию.

Интерфейс программы выглядит следующим образом:



Интерфейс ROSA Memory Clean

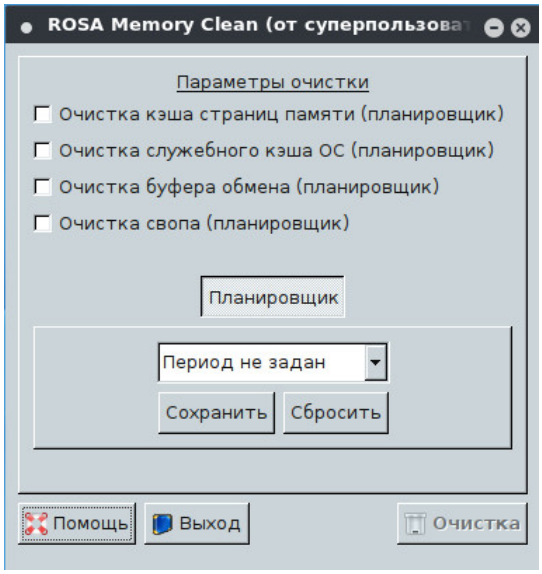
5.3.2. Описание элементов интерфейса

В верхней части окна перечислены различные варианты очистки памяти:

1) **Очистка кэша страниц памяти**. Соответствующие страницы памяти получаются в результате чтения и записи обычных файлов на файловых системах, специальных файлов блочных устройств и файлов, отображаемых в память. Таким образом, в страничном кэше содержатся страницы памяти, полностью заполненные данными из файлов, к которым только что производился доступ.
2) **Очистка служебного кэша ОС** — удаление различных служебных элементов работы ОС, например, так называемых элементов каталога. Данные объекты создаются «на лету» на основании строкового представления имени пути к конкретному файлу в результате внутреннего перевода системой элементов пути. Также удаляются индексные дескрипторы. Это структуры, хранящие метаинформацию о файлах, каталогах или других объектах файловой системы.
3) **Очистка буфера обмена** — освобождение промежуточного хранилища данных, служащего для их переноса между приложениями или в рамках одного приложения.

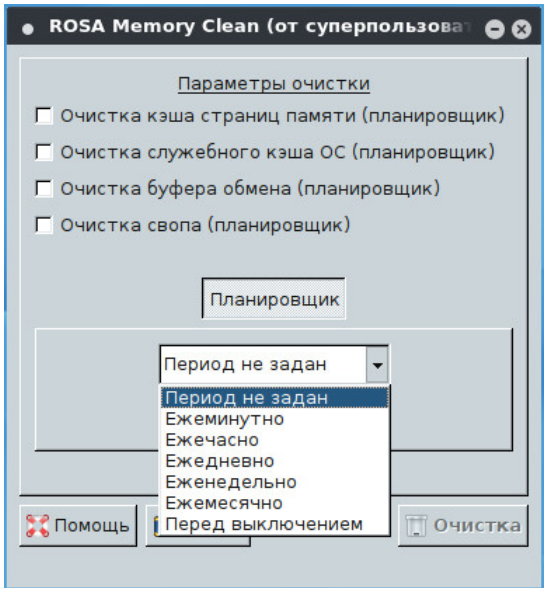
4) **Очистка свопа** — перезапуск механизма виртуальной памяти, перемещающего фрагменты данных из оперативной памяти в хранилище (например, жесткий диск или внешний флеш-накопитель).

Кнопка [Планировщик] позволяет перейти в режим планирования очистки по расписанию. Внешний вид окна при нажатии на кнопку меняется:



Внешний вид ROSA Memory Clean при переходе в режим планировщика

При нажатой кнопке [Планировщик] появятся несколько новых элементов интерфейса — кнопки [Сохранить], [Сбросить] и выпадающий список с выбором периода очистки:

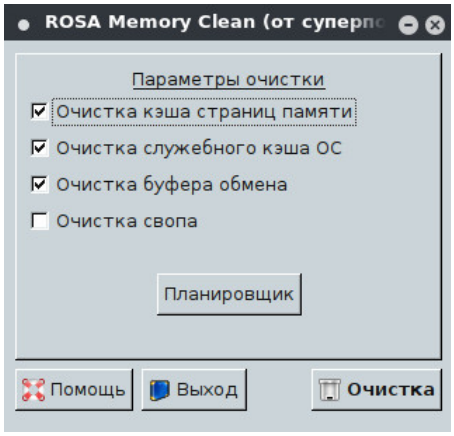


Выбор периода очистки

Повторное нажатие кнопки [Планировщик] скроет окошко с периодами и кнопками управления планированием очистки. Кнопка [Помощь] открывает данный документ. По кнопке [Выход] происходит выход из программы. Нажатие кнопки [Очистка] при выбранных параметрах после подтверждения запустит процесс очистки.

5.3.3. Работа с программой

Для запуска процесса очистки памяти необходимо выбрать один или несколько из представленных параметров, после чего нажать на кнопку [Очистка].



Выбор областей памяти для очистки

После нажатия на кнопку [Очистка] нужно будет подтвердить выбранные действия. При положительном ответе запустится процесс очистки и появится окно ожидания операции. Необходимо дождаться ее окончания. По окончании окно ожидания исчезнет, и в нижней части появится сообщение «Операция завершена».

Очистку памяти можно сделать периодической, чтобы не запускать программу каждый раз вручную. Для этого нужно нажать на кнопку [Планировщик]. Выбрав период очистки, нажмите на кнопку [Сохранить], чтобы записать период в конфигурационный файл программы. Теперь очистка памяти будет запускаться автоматически без дополнительных действий пользователя. Нажатие кнопки [Сбросить] удалит выбранные ранее период и параметры очистки.

5.4. Использование утилиты ROSA Shred

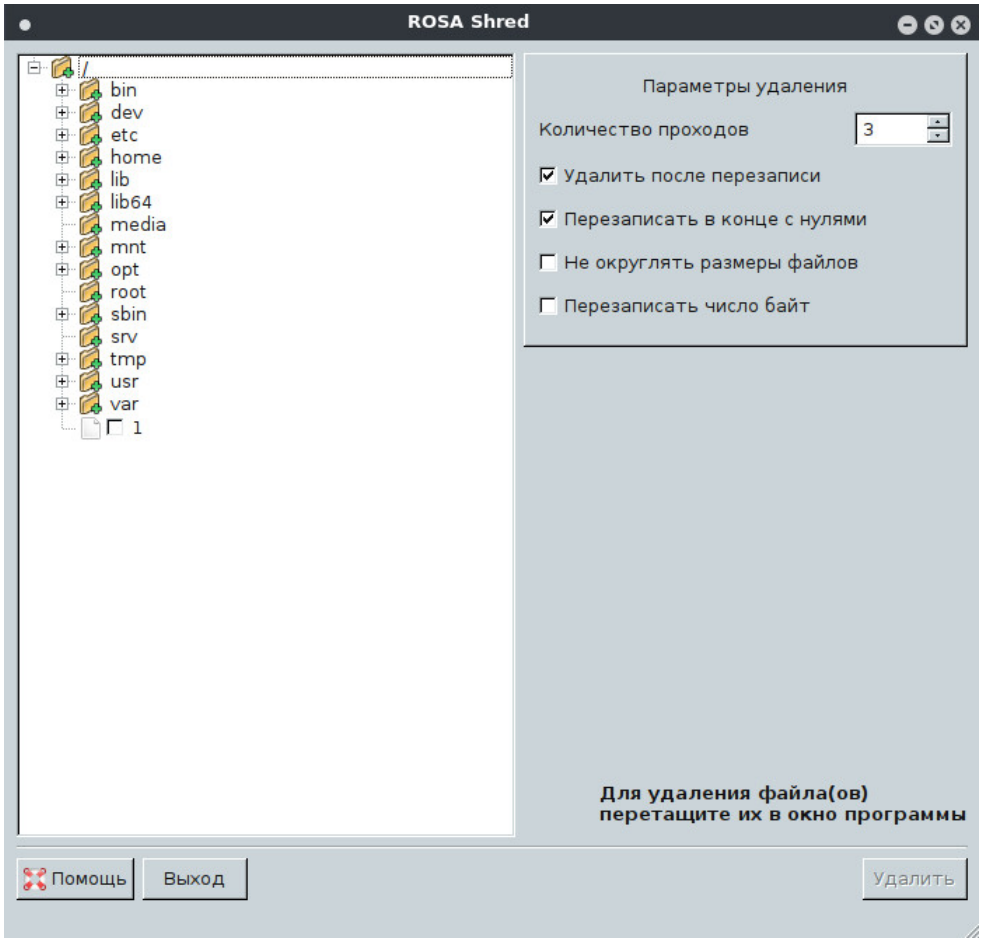
Утилита работает только от имени пользователя, под учетной записью которого она была запущена. Таким образом, она сможет удалить лишь те файлы, на которые у данного пользователя есть права, достаточные для удаления файла.

5.4.1. Введение

Утилита ROSA Shred несколько раз перезаписывает указанные файлы специальными битовыми последовательностями, чтобы сделать более трудоемким процесс восстановления данных даже в случае использования специального оборудования. Используйте ее для надежного удаления конфиденциальной информации.

Чтобы запустить утилиту, выберите пункт меню «Приложения → Утилиты СЗИ ОС РОСА → ROSA Shred» или выполните команду `rosa-shred`.

Интерфейс программы выглядит следующим образом:



Интерфейс ROSA Shred

В окне слева представлена корневая файловая система вашей ОС. Справа представлены опции удаления, которые можно установить для файлов.

5.4.2. Опции перезаписи

1) Количество проходов=N. Переписать N раз вместо указанных (3) по умолчанию.
2) Перезаписать в конце с нулями. Перезаписать в конце с нулями, чтобы скрыть перемешивание.
3) Удалить после перезаписи. Перемешать и удалить файл после перезаписи.
4) Не округлять размеры файлов. Не округлять размеры файлов до следующего целого блока; по умолчанию для нерегулярных файлов.
5) Перезаписать количество байт=N. Очистить N байт, а не файл целиком.

6. УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

Оператор, которому предстоит работать с ОС РОСА «КОБАЛЬТ», должен быть зарегистрирован в ней как пользователь. Ни один пользователь не может начать работу с ОС, не идентифицировав себя и не предоставив аутентификационную информацию, подтверждающую, что пользователь действительно является тем, за кого себя выдает.

6.1. Аутентификация и идентификация

Механизмы идентификации и аутентификации являются обязательными компонентами модели защиты. Каждый пользователь имеет свой уникальный числовой идентификатор (UID) — положительное целое число, которое обычно выбирается автоматически при регистрации учетной записи. Это число не может быть произвольным, поскольку в ОС существуют правила, определяющие, каким типам пользователей могут быть выданы идентификаторы из того или иного диапазона.

Идентификатору пользователя соответствует системное имя пользователя (учетная запись). Для привилегированного пользователя с учетной записью root зарезервирован нулевой идентификатор.

Для более удобного управления доступом к ресурсам все пользователи включаются в группы. Группа — это подмножество пользователей, объединенных по каким-либо критериям. У группы, так же, как и у пользователя, есть имя и идентификационный номер — GID. В ОС каждый пользователь должен принадлежать как минимум к одной группе — группе по умолчанию. При создании учетной записи пользователя обычно создается и группа, имя которой совпадает с системным именем пользователя. Именно эта группа будет использоваться как группа по умолчанию для данного пользователя. Максимальное количество групп, в которых может состоять один пользователь, равно 32.

Аутентификация — это процесс установления подлинности пользователя. Для аутентификации в ОС РОСА «КОБАЛЬТ» используется пароль. Пароль представляет собой набор символов, известный только его владельцу и используемый для удостоверения его подлинности. Каждый пользователь имеет собственный пароль. Наличие пароля — необходимая составляющая политики безопасности пользователей. Без пароля, зная только имя пользователя, осуществить вход невозможно.

Имена учетных записей пользователей и их идентификаторы хранятся в файле /etc/passwd. Право на запись в этот файл имеет только пользователь root; читать этот файл могут все пользователи. Каждая запись в нем содержит следующие поля:

1) Системное имя пользователя.
2) Признак пароля. Символ `x` обозначает наличие пароля.
3) Идентификатор пользователя (UID). Каждый пользователь имеет свой уникальный идентификационный номер, который используется, например, при установке прав доступа на файлы.
4) Идентификатор основной группы пользователя (GID). В этом поле указывается

идентификационный номер группы, к которой принадлежит пользователь.

5) Дополнительная информация (GECOS). Используется (опционально) для хранения дополнительной информации о пользователе, например, его полного имени.
6) Путь к домашнему каталогу пользователя.
7) Путь к командной оболочке. Содержит полный путь к рабочей оболочке пользователя (по умолчанию такой оболочкой является bash). Эта оболочка запускается, когда пользователь проходит процедуру аутентификации.

6.2. Команды для управления пользователями

Для управления пользователями, как правило, используются следующие команды:

- `id` — показывает ID пользователей и групп;
- `useradd`, `adduser`, `usermod`, `userdel` — стандартные утилиты для добавления, изменения и удаления учетных записей пользователей;
- `groupadd`, `groupmod`, `groupdel` — стандартные утилиты для добавления, изменения и удаления групп;
- `gpasswd` — утилита, служащая в основном для изменения паролей групп в файле /etc/gshadow, который используется командой newgrp;
- `pwck`, `grpck` — утилиты, которые можно использовать для проверки паролей, групп и связанных с ними теневых файлов;
- `pwconv`, `pwunconv` — утилиты, которые можно использовать для преобразования стандартных паролей в теневые пароли и наоборот;
- `grpconv`, `grpunconv` — по аналогии с предыдущими программами, эти утилиты можно использовать для преобразования теневой информации для групповых учетных записей.

6.3. Добавление нового пользователя

Чтобы добавить в систему нового пользователя, выполните следующую команду с привилегиями суперпользователя root:

```
# useradd [параметры] username
```

Параметры команды useradd:

- `-c <комментарий>`. Комментарий может быть любой строкой. Этот параметр обычно используется для указания полного имени пользователя;
- `-d <домашний_каталог>`. Создание домашнего каталога вместо каталога по умолчанию, т. е. `/home/<имя_пользователя>/`;
- `-e <дата>`. Дата отключения учетной записи в формате ГГГГ-ММ-ДД;
- `-f <дней>`. Число дней от момента истечения срока действия пароля и до момента отключения учетной записи. Если был указан `0`, учетная запись отключается сразу же после истечения срока действия пароля. Если было указано число `-1`, после истечения срока действия пароля учетная запись пользователя не отключается;

- `-g <имя_группы>`. Имя или номер для группы пользователя по умолчанию (первичной группы). На момент создания пользователя группа должна уже существовать;
- `-G <список_групп>`. Список дополнительных (добавочных, отличных от группы по умолчанию) имен или номеров групп, в которых состоит пользователь, разделенных запятой. Эти группы должны существовать к моменту создания пользователя;
- `-m`. Создать домашний каталог при его отсутствии;
- `-M`. Не создавать домашний каталог;
- `-N`. Не создавать частную группу для пользователя;
- `-p <пароль>`. Пароль, зашифрованный с помощью *crypt*;
- `-r`. Создать системную учетную запись со значением UID меньше 1000 и без домашнего каталога;
- `-s`. Командный интерпретатор пользователя, по умолчанию /bin/bash;
- `-u <uid>`. Пользовательский ID, значение должно быть уникальным и не превышать 999.

Примечание. В ОС РОСА «КОБАЛЬТ» для системных пользователей по умолчанию используется диапазон ID 1–999. Диапазоны UID и GID по умолчанию можно изменить в файле /etc/login.defs.

По умолчанию команда `useradd` создает заблокированную учетную запись пользователя. Чтобы разблокировать учетную запись, выполните следующую команду для присвоения пароля:

```
# passwd <имя_пользователя>
```

Примечание. При использовании команды `adduser <имя_пользователя>` домашний каталог создается автоматически при первом входе пользователя.

6.4. Добавление новой группы

Имена групп и их идентификаторы хранятся в файле /etc/group. Каждая запись в этом файле содержит следующие поля:

1) Имя группы.
2) Признак пароля. Обычно не используется.
3) Идентификатор группы (GID).
4) Члены группы. В этом поле перечисляются пользователи, для которых группа является дополнительной.

Чтобы добавить в систему новую группу, выполните следующую команду с привилегиями суперпользователя root:

```
# groupadd [<параметры>] <имя_группы>
```

Параметры команды `groupadd`:

- `-f`, `--force`. В сочетании с ключом `-g <gid>`, если этот GID уже существует, `groupadd` выберет для группы другой уникальный GID;

- `-g <gid>`. Идентификатор группы. Должен иметь уникальное значение, превышающее 999;
- `-K`, `--key <ключ>=<значение>`. Перезаписать значения по умолчанию для файла /etc/login.defs;
- `-o`, `--non-unique`. Разрешить создание групп с дублирующимися GID;
- `-p`, `--password пароль`. Использовать для группы этот зашифрованный пароль;
- `-r`. Создать системную группу со значением GID менее 1000.

6.5. Добавление существующего пользователя в существующую группу

Для добавления существующего пользователя в существующую группу используется команда `usermod`. Различные параметры `usermod` по-разному влияют на первичную группу пользователя и дополнительные группы пользователя.

Для изменения первичной группы пользователя выполните следующую команду с привилегиями суперпользователя root:

```
# usermod -g <имя_группы> <имя_пользователя>
```

Для изменения дополнительных групп пользователя выполните следующую команду:

```
# usermod -G <имя_группы_1>, <имя_группы_2>, ... <имя_пользователя>
```

Обратите внимание, что в этом случае все ранее существовавшие параметры дополнительных групп пользователя будут заменены на другую группу или несколько других групп.

Чтобы добавить одну или несколько групп к дополнительным группам пользователя, выполните одну из следующих команд:

```
# usermod -aG <имена_групп> <имя_пользователя>
# usermod --append -G <имена_групп> <имя_пользователя>
```

Обратите внимание, что в этом случае другие группы добавляются к дополнительным группам пользователя.

6.6. Удаление пользователя

Чтобы удалить пользователя, выполните следующую команду с привилегиями суперпользователя root:

```
# userdel [<параметры>] <имя_пользователя>
```

Параметры команды `userdel`:

- `-r`, `--remove`. Удалить также файлы пользователя (содержимое домашнего каталога, электронную почту и т. д.).

6.7. Создание каталогов групп

Для каждого крупного проекта системные администраторы обычно создают группу и присваивают этим группам пользователей для предоставления им доступа к файлам этого проекта. В рамках этой традиционной схемы управление файлами является трудной зада-

чей; если кто-нибудь создает файл, он связывается с первичной группой, к которой принадлежат пользователи. Если один и тот же человек работает в нескольких проектах, становится трудно привязать правильные файлы к соответствующей группе. Тем не менее, в рамках схемы UPG группы автоматически присваиваются файлам, созданным в каталоге, для которого настроен бит setgid. Бит setgid существенно упрощает управление групповыми проектами с общими каталогами, т. к. любые файлы, создаваемые пользователем в каталоге, принадлежат группе, которой принадлежит этот каталог.

Предположим, что группе сотрудников нужно работать с файлами в каталоге /opt/myproject/. Некоторым сотрудникам разрешено изменять содержимое этого каталога, но не всем.

1) Создайте каталог /opt/myproject/, выполнив следующую команду с привилегиями суперпользователя root:
   ```
   # mkdir /opt/myproject
   ```
2) Добавьте группу myproject в систему:
   ```
   groupadd myproject
   ```
3) Свяжите содержимое каталога /opt/myproject/ с группой myproject:
   ```
   chown root:myproject /opt/myproject
   ```
4) Разрешите пользователям в группе создавать файлы в этом каталоге и установите бит setgid:
   ```
   chmod 2775 /opt/myproject
   ```

На этом этапе все члены группы myproject могут создавать и изменять файлы в каталоге /opt/myproject/, так что системному администратору нет необходимости изменять права доступа на файлы каждый раз, когда пользователи создают новые файлы.

5) Чтобы проверить, что права на файлы были установлены правильно, выполните следующую команду:
   ```
   # ls -ld /opt/myproject
   drwxrwsr-x. 3 root myproject 4096 Mar 3 18:31 /opt/myproject
   ```
6) Добавьте пользователей в группу myproject:
   ```
   usermod -aG myproject username
   ```

6.8. Установление прав доступа по умолчанию для новых файлов с помощью umask

Когда процесс создает файл, у этого файла есть определенные права доступа по умолчанию, например, `-rw-rw-r--`. Эти начальные права доступа частично определяются маской режима создания файлов, также называемой маской прав файлов или umask. У каждого процесса есть своя umask, например, umask для bash по умолчанию имеет значение 0022. Значение umask процесса можно изменять.

6.8.1. Управление значениями umask в командных интерпретаторах

В популярных командных интерпретаторах, таких как bash, ksh, zsh и tcsh, значение umask регулируется umask, встроенной в командный интерпретатор. Процесс, запущенный в командном интерпретаторе, наследует его umask.

6.8.2. Просмотр текущей маски

Чтобы посмотреть текущее значение umask в восьмеричном формате, выполните:

```
$ umask
0022
```

Чтобы посмотреть текущее значение umask в символьном формате, выполните:

```
$ umask -S
u=rwx,g=rx,o=rx
```

6.8.3. Установка маски в командном интерпретаторе с помощью umask

Чтобы настроить значение umask для текущего сеанса командного интерпретатора, выполните:

```
$ umask <восьмеричная_маска>
```

Замените параметр `<восьмеричная_маска>` четырьмя цифрами от 0 до 7. При указании трех или менее цифр права доступа устанавливаются так, как если бы в начале были указаны нули. Например, umask 7 интерпретируется как 0007.

6.8.4. Установка значения umask в восьмеричном формате

Чтобы запретить владельцу и группе запись и выполнение вновь создаваемых файлов, а также любые права доступа для остальных, выполните:

```
$ umask 0337
```

Чтобы установить значение umask для текущего сеанса командного интерпретатора в символьном формате, выполните:

```
$ umask -S <символьная_маска>
```

6.8.5. Установка значения umask в символьном формате

Чтобы установить значение umask 0337 в символьном формате, выполните:

```
$ umask -S u=r,g=r,o=
```

6.8.6. Работа со значением umask в командном интерпретаторе по умолчанию

Значение umask командного интерпретатора обычно записывается в его конфигурационном файле. Для bash это /etc/bashrc. Чтобы просмотреть значение umask по умолчанию для bash, выполните:

```
$ grep -i -B 1 umask /etc/bashrc
```

Посмотреть, назначено ли значение umask, можно либо с помощью команды `umask`, либо с помощью переменной UMASK. В примере ниже umask настроена на значение 022 с помощью команды `umask`:

```
$ grep -i -B 1 umask /etc/bashrc
   # By default, we want umask to get set. This sets it for non-
login shell.
--
   if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
```

```
 umask 002
else
 umask 022
```

Чтобы сменить значение umask по умолчанию для bash, смените вызов команды umask или присвоение переменной UMASK в файле /etc/bashrc. Пример ниже показывает, как сменить umask по умолчанию на значение 0227:

```
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
 umask 002
else
 umask 227
```

6.8.7. Управление значением umask в командном интерпретаторе по умолчанию для конкретного пользователя

По умолчанию значение umask в bask для нового пользователя определяется в /etc/bashrc.

Чтобы сменить значение umask для bash для конкретного пользователя, добавьте вызов для команды umask в файле $HOME/.bashrc для этого пользователя. Чтобы, например, сменить umask для пользователя ivanov на 0227, выполните:

```
ivanov@server ~]$ echo 'umask 227' >> /home/ivanov/.bashrc
```

6.8.8. Изменение прав доступа по умолчанию для создаваемых домашних каталогов

Чтобы изменить права доступа, с которыми создаются домашние каталоги пользователей, измените значение переменной UMASK в файле /etc/login.defs:

```
# The permission mask is initialized to this value. If not
specified,
# the permission mask will be initialized to 022.
UMASK 077
```

Дополнительную информацию о различных утилитах для управления пользователями и группами см. на man-страницах:

- useradd(8) — команда `useradd`: как создавать новых пользователей;
- userdel(8) — команда `userdel`: как удалять пользователей;
- usermod(8) — команда `usermod`: как изменять параметры пользователей;
- groupadd(8) — команда `groupadd`: как создавать новые группы;
- groupdel(8) — команда `groupdel`: как удалять группы;
- groupmod(8) — команда `groupmod`: как изменять членство в группах;
- gpasswd(1) — команда `gpasswd`: как работать с файлом /etc/group;
- grpck(8) — команда `grpck`: как проверить целостность файла /etc/group;
- pwck(8) — команда `pwck`: как проверить целостность файлов /etc/passwd и /etc/shadow;

- pwconv(8) — команды `pwconv`, `pwunconv`, `grpconv` и `grpunconv`: как преобразовывать теневую информацию для пользователей и групп;
- id(1) — команда `id`: как просмотреть идентификаторы пользователей и групп;
- umask(2) — команда `umask`: как работать с маской режима создания файлов.

Информацию о соответствующих файлах конфигурации см. в следующих man-страницах:

- group(5) — как использовать файл /etc/group для настройки системных групп;
- passwd(5) — как использовать файл /etc/passwd для настройки информации о пользователях;
- shadow(5) — как использовать файл /etc/shadow для настройки паролей и информации о сроках действия учетных записей в системе.

6.9. Защита паролей

В целях безопасности во время установки ОС РОСА «КОБАЛЬТ» система настраивается на использование алгоритма безопасного хеширования 512 (SHA512) и теневых паролей. Крайне рекомендуем не изменять этих параметры.

Если во время установки не выбрать параметр теневых паролей, все пароли будут храниться в однонаправленном хеше в файле /etc/passwd, открытом для чтения для всех, что делает систему уязвимой для внешних атак, направленных на взлом пароля. Если недоброжелатель сможет получить доступ к машине под учетной записью обычного пользователя, он сможет скопировать файл /etc/passwd на свою машину и запустить любые программы по взлому пароля. Если в файле будет ненадежный (слабый) пароль, то его обнаружение недоброжелателем — это лишь вопрос времени.

Теневые пароли исключают такой тип атак, т. к. хеши паролей хранятся в файле /etc/shadow, доступ на чтение которого есть только у суперпользователя root.

6.9.1. Файл /etc/shadow

Для генерации файла /etc/shadow используется утилита *pwconv*, а для отказа от использования этого файла — *pwunconv*. Для изменения пароля используется утилита *passwd*, которая не дает установить легко взламываемый пароль. В качестве параметра в командной строке она получает имя пользователя и при запуске требует ввода пароля для этого пользователя. После подтверждения пароль шифруется и сохраняется в файле /etc/shadow.

Записи в файле /etc/shadow состоят из нескольких полей, разделенных символами «:».

1) Имя пользователя. Это поле дублируется из файла /etc/passwd.
2) Хеш пароля. Если данное поле содержит знак `!` или `*`, это означает, что учетная запись заблокирована и пользователь не сможет осуществить вход. Если поле содержит `!!`, это означает, что у пользователя никогда не было пароля и, не назначив его, он не сможет осуществить вход.
3) Дата последней смены пароля. В этом поле записывается число дней, прошедших с

1 января 1970 г. до даты, когда пользователь сменил пароль в последний раз. Эта информация используется вместе со следующими полями, управляющими сроком действия пароля.

4) Число дней, которое должно пройти до смены пароля. Минимальный срок (в днях), который должен истечь, прежде чем пользователь сможет сменить пароль.
5) Число дней, после которого необходимо сменить пароль. Максимальный срок (в днях), по истечении которого необходимо сменить пароль.
6) Число дней до предупреждения о необходимости смены пароля. Число дней до истечения срока действия пароля, в течение которых пользователь будет получать предупреждения о скором окончании срока действия пароля.
7) Число дней до отключения учетной записи. Число дней, которое должно пройти с момента окончания срока действия пароля до отключения учетной записи.
8) Дата блокировки учетной записи. Дата (в днях, прошедших с 1 января 1970 г.), когда учетная запись пользователя будет (или была) отключена.
9) Зарезервированное поле. Это поле игнорируется.

Пример строки файла /etc/shadow:

```
tester:$1$.QKDPc5E$SWlkjRWexrXYgc98F.:17666:0:60:7:10:18031:
```

- пароль был в последний раз изменен 15 мая 2018 г.;
- срок, в течение которого нельзя изменить пароль, не определен;
- пароль должен меняться каждые 60 дней;
- пользователь будет получать предупреждение о необходимости его сменить в течение 7 дней;
- учетная запись будет отключена через 10 дней после истечения срока действия пароля, если не будет попыток входа;
- срок действия учетной записи истекает 15 мая 2019 г.

Для изменения временн*ых* параметров учетной записи пользователя используется утилита *chage*. Чтобы получить подробную информацию по ее использованию, выполните команду `man chage`.

Для проверки синтаксиса файлов паролей используются утилиты *pwck* и *grpck*.

Утилита *pwck* последовательно анализирует записи файлов /etc/passwd и /etc/shadow, проверяя, что каждая запись содержит:

- правильное количество полей;
- уникальное имя пользователя;
- действительные идентификаторы пользователей и групп;
- действительную первичную группу;
- действительный домашний каталог;
- действительный командный процессор.

Утилита grpck выполняет проверку файлов /etc/group и /etc/gshadow. Она последовательно анализирует записи файлов и проверяет, что каждая запись содержит:

- правильное количество полей;
- уникальное имя группы;

- действительный список членов и администраторов группы.

Использование теневых паролей вынуждает недоброжелателя осуществлять попытку удаленного взлома пароля через авторизацию в сетевом сервисе на машине, таком, как SSH или FTP. Подобный тип атак с полным перебором (брутфорс) выполняется гораздо медленнее и оставляет очевидный след в виде сотен неудачных попыток авторизации, записанных в системный журнал. Конечно, если взломщик начнет атаку на систему со слабым паролем в полночь, то к рассвету он уже может получить доступ и изменить файлы журнала для скрытия следов.

Кроме формата и условий хранения паролей, также существует проблема содержимого. Единственное, что пользователь может сделать, чтобы защитить свою учетную запись от попыток взлома пароля, — это придумать надежный пароль.

6.9.2. Принудительное создание надежных паролей

В организациях с большим количеством пользователей у системных администраторов есть два возможных пути для принуждения пользователей к созданию надежных паролей. Они могут создать пароль для пользователя самостоятельно, или же дать пользователю самому создать пароль с проверкой на его соответствие требованиям безопасности.

Создание паролей для пользователей гарантирует надежность паролей, но становится тяжелой обязанностью по мере роста организации. Также это провоцирует пользователей записывать свои пароли, что повышает риск. В силу этих причин, большинство системных администраторов предпочитают, чтобы пользователи сами создавали свои пароли, но подвергают эти пароли строгим проверкам.

Требования к паролю:

- секретность;
- устойчивость к угадыванию;
- устойчивость к атаке перебором.

В некоторых случаях администраторы могут заставлять пользователей менять пароли с помощью установки срока действия пароля. В конце срока действия пароля пользователю будет предложено ввести новый пароль, который затем может быть использоваться, пока срок его действия также не закончится. При выборе срока действия пароля следует соблюдать разумный баланс между безопасностью и удобством для пользователей. Также необходимо вести историю ранее использованных паролей, чтобы избежать их повторения в будущем.

При попытке входа набранный пароль снова шифруется и сравнивается с записью в файле /etc/shadow, хранящем хеши паролей. Совпадение означает, что пароль введен верно, и доступ к изделию разрешается.

При необходимости сменить пароль пользователи могут использовать консольную утилиту *passwd*, поддерживающую протокол PAM (Pluggable Authentication Modules) и способную проверить пароль на необходимую длину и другие факторы устойчивости к взлому. Эту проверку выполняет модуль PAM pam_pwquality.so.

Примечание. В ОС РОСА «КОБАЛЬТ» модуль pam_pwquality пришел на смену модулю pam_cracklib, который использовался ранее в подобных Linux-системах по умолча-

нию для проверок паролей на безопасность.

Для проверок паролей на соответствие требованиям безопасности модуль pam_pwquality использует набор правил. Процедура проверки состоит из двух шагов: сначала выполняется проверка на наличие слова в словаре. Если слово не встретилось, далее выполняется некоторое количество дополнительных проверок. Модуль pam_pwquality наряду с другими модулями PAM используется в качестве компонента паролей в файле /etc/pam.d/passw, а набор правил указывается в конфигурационном файле /etc/security/pwquality.conf. Полный список этих проверок см. на man-странице pwquality.conf (8).

6.9.3. Настройка проверки паролей на безопасность в файле pwquality.conf

Чтобы активировать использование pam_quality, добавьте следующую строку в файл /etc/pam.d/passwd:

```
password    required pam_pwquality.so retry=3
```

Параметры проверки указываются по одному на строку. Чтобы, например, пароль имел не менее 8 символов в длину и включал в себя все четыре класса символов, добавьте следующие строки в файл /etc/security/pwquality.conf:

```
minlen = 8
minclass = 4
```

Чтобы настроить проверку на последовательность символов и последовательный повтор одних и тех же символов, добавьте следующие строки в файл /etc/security/pwquality.conf:

```
maxsequence = 3
maxrepeat = 3
```

В этом примере введенный пароль не может содержать более 3 символов в монотонной последовательности (например, «абвгд»), и более 3 идентичных последовательных символов («1111»).

Примечание. Поскольку пользователь root является тем, кто принуждает к использованию правил соответствия паролей требованиям безопасности, он может создавать любой пароль как для своей, так и для пользовательской учетной записи, игнорируя предупреждения безопасности.

6.9.4. Настойка сроков действия паролей

Ограничение сроков действия паролей — еще один способ, используемый системными администраторами для защиты от слабых паролей в подотчетной организации. Срок действия пароля означает, что по истечении указанного времени (обычно это 90 дней) пользователю будет показан запрос о создании нового пароля. В теории, если пользователь будет вынужден периодически менять пароль, возможный взломанный пароль будет полезен недоброжелателю только в течение ограниченного промежутка времени. Минус принудительной смены паролей в том, что пользователи, скорее всего, будут свои пароли где-то записывать.

Для указания сроков принудительной смены паролей используйте команду `chage`.

Примечание. В ОС РОСА «КОБАЛЬТ» теневые пароли используются по умолчанию.

Параметр `-M` команды `chage` указывает максимальное число дней, в течение которых пароль остается действительным. Чтобы указать, что срок действия пароля пользователя закончится через 90 дней, выполните следующую команду:

```
chage -M 90 <имя_пользователя>
```

Замените в вышеприведенной команде `<имя_пользователя>` системным именем пользователя. Чтобы отключить срок действия пароля, после параметра `-M` укажите параметр `-1`.

С командой `chage` можно также работать в интерактивном режиме для изменения нескольких сроков действия паролей и параметров учетных записей. Чтобы войти в интерактивный режим, выполните:

```
chage <имя_пользователя>
```

Ниже приведен пример интерактивной сессии использования этой команды:

```
# chage vasily

Changing the aging information for vasily
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
```

Можно настроить смену пароля сразу же после первой авторизации пользователя в системе, чтобы принудить пользователя сменить пароль немедленно.

Настройте начальный пароль. Чтобы присвоить пароль по умолчанию, выполните следующую команду с привилегиями суперпользователя root:

```
# passwd <имя_пользователя>
```

Примечание. Утилита passwd предоставляет возможность настроить нулевой пароль. Это удобно, но крайне ненадежно, так любое постороннее лицо может войти в систему и получить доступ с использованием такого незащищенного пользователя. При любой возможности избегайте использования нулевых паролей. Если это невозможно, всегда сначала нужно убедиться, что пользователь уже готов войти в систему, перед тем, как разблокировать учетную запись с нулевым паролем.

Команда для немедленной принудительной смены пароля (с правами root):

```
# chage -d 0 пользователь
```

Эта команда настраивает значение последней смены пароля на начало отсчета времени (1 января 1970 года). Это значение принуждает к немедленному прекращению сроку действия пароля вне зависимости от параметров политики срока действия паролей, если такая существует. После первой авторизации в системе пользователю выводится запрос на указание нового пароля.

6.9.4.1. Описание полей файла конфигурации качества паролей

Параметр, указывающий, какое количество символов нового пароля не должно совпадать с символами старого пароля:

```
difok = 5
```

Минимальное количество символов в пароле:

```
minlen = 9
```

Класс устойчивости пароля 1 — минимальное количество цифр в пароле:

```
dcredit = 1
```

Класс устойчивости пароля 2 — минимальное количество заглавных букв в пароле:

```
ucredit = 1
```

Класс устойчивости пароля 3 — максимальное количество строчных букв в пароле:

```
lcredit = 1
```

Класс устойчивости пароля 4 — минимальное количество спецсимволов в пароле:

```
ocredit = 1
```

Количество учитываемых классов для качества пароля (число от 1 до 4):

```
minclass = 0
```

Максимально разрешенное последовательное количество повторяемых символов нового пароля по отношению к старому:

```
maxrepeat = 0
```

Максимально разрешенное последовательное количество повторяемых символов нового пароля по отношению к старому с учетом класса качества пароля:

```
maxclassrepeat = 0
```

Проверять ли пароль на соответствие данным, содержащимся в поле GECOS (имя, иные необязательные данные пользователя, ФИО, номер телефона). Если указана единица — проверка производится, если указан 0 — нет:

```
gecoscheck = 0
```

Путь к словарю. Указывается текстовый файл, с которым производится сверка паролей. Иначе сверка не производится:

```
dictpath =
```

6.10. Блокировка учетных записей пользователей

В ОС РОСА «КОБАЛЬТ» модуль PAM pam_faillock дает системным администраторам возможность заблокировать учетную запись пользователя после нескольких неудачных попыток авторизации. Ограничение числа попыток авторизации пользователей служит в основном средством защиты от возможных атак типа брутфорс, направленных на получение пароля учетной записи пользователя.

При наличии модуля pam_faillock неудачные попытки авторизации записываются в отдельный файл для каждого пользователя в каталоге /var/run/faillock/.

Примечание. Порядок строк в неудачных попытках авторизации очень важен. Любая попытка сменить этот порядок может заблокировать учетные записи всех пользователей, включая суперпользователя root, если используется параметр `even_deny_root`.

Для настройки блокировки учетных записей выполните следующие действия:

1) Проверьте, являются ли файлы system-auth и password-auth символьными ссылками, указывающими на system-auth-ac и password-auth-ac (это параметры по умолчанию):

```
# ls -l /etc/pam.d/{password,system}-auth
```

Если вывод команды будет аналогичен следующему, значит, символьные ссылки на месте, и можно переходить к шагу 3:

```
lrwxrwxrwx. 1 root root 16 24. Feb 09.29 /etc/pam.d/password-auth
-> password-auth-ac
lrwxrwxrwx. 1 root root 28 24. Feb 09.29 /etc/pam.d/system-auth
-> system-auth-ac
```

Если файлы system-auth и password-auth не являются символьными ссылками, переходите к следующему шагу.

2) Переименуйте конфигурационные файлы:

```
# mv /etc/pam.d/system-auth /etc/pam.d/system-auth-ac
# mv /etc/pam.d/password-auth /etc/pam.d/password-auth-ac
```

3) Создайте файлы с конкретными параметрами:

```
# vi /etc/pam.d/system-auth-local
Файл /etc/pam.d/system-auth-local должен содержать следующие
строки:
auth     required pam_faillock.so preauth silent audit deny=3
unlock_time=600
auth     include system-auth-ac
auth     [default=die] pam_faillock.so authfail silent audit
deny=3 unlock_time=600

account     required pam_faillock.so
account     include system-auth-ac
password    include system-auth-ac
session     include system-auth-ac

# vi /etc/pam.d/password-auth-local
```

Файл /etc/pam.d/password-auth-local должен содержать следующие строки:

```
auth     required pam_faillock.so preauth silent audit deny=3
unlock_time=600
auth     include password-auth-ac
auth     [default=die] pam_faillock.so authfail silent audit
deny=3 unlock_time=600

account     required pam_faillock.so
account     include password-auth-ac
password    include password-auth-ac
session     include password-auth-ac
```

4) Создайте следующие символьные ссылки:

```
# ln -sf /etc/pam.d/system-auth-local /etc/pam.d/system-auth
# ln -sf /etc/pam.d/password-auth-local /etc/pam.d/password-auth
```

Подробности о возможных параметрах pam_faillock см. на man-странице pam_faillock(8).

6.11. Ограничения на вход локального пользователя ОС

6.11.1. Запрет входа в ОС после нескольких неправильных попыток ввода пароля

Для настройки ограничения по неуспешным попыткам доступа необходимо добавить в файл /etc/pam.d/system-auth и /etc/pam.d/password-auth две строки:

- в секцию `auth` (самой верхней строкой):

```
auth required pam_tally2.so file=/var/log/tallylog deny=3
even_deny_root unlock_time=1200
```

Здесь `deny=3` задает количество неуспешных попыток входа пользователя до его блокировки, а `unlock_time` — время (в секундах), на которое блокируется пользователь. Если не указывать данный параметр, разблокировать пользователя можно будет только вручную с помощью суперпользователя root;

- в секцию `account` (после строки `account required pam_unix.so`)

```
account required pam_tally2.so
```

Посмотреть данные по неуспешным входам пользователей можно с помощью команды `pam_tally2`.

6.11.2. Запрет входа в ОС в определенное время

Чтобы запретить запуск пользователем любых программ (включая вход в ОС) в будние дни с 18:00 до 9:00, выполните следующие действия:

1) Добавьте в файл /etc/security/time.conf следующую строку:

```
*;*;user1;!Al1800-2400|Al0000-9000
```

2) Укажите в файле /etc/pam.d/password-auth самым первым правилом в строчках `account` следующее:

```
account required pam_time.so
```

6.12. Блокировка виртуальных текстовых консолей с помощью утилиты vlock

С помощью утилиты `vlock` пользователи могут заблокировать виртуальную консоль. В ОС РОСА «КОБАЛЬТ» данная утилита устанавливается по умолчанию. Разблокировать консоль может пользователь — владелец консоли либо администратор root.

Любой консольный сеанс можно заблокировать путем выполнения команды `vlock` без дополнительных параметров. Текущий сеанс виртуальной консоли будет заблокиро-

ван без прерывания доступа других пользователей. Чтобы предотвратить доступ ко всем виртуальным консолям машины, выполните:

```
vlock -a
```

В этом случае `vlock` блокирует текущую активную консоль, а параметр `-a` предотвращает переключение в другие виртуальные консоли. Подробности см. на man-странице vlock(1).

6.13. Блокирование входа для системных пользователей

В целях безопасности для системных пользователей в поле «Путь к командной оболочке» файла /etc/passwd рекомендуется указывать /sbin/nologin. Сама по себе программа *nologin* не является оболочкой, единственное ее назначение — не допустить вход в ОС. При попытке входа под именем пользователя, у которого в качестве рабочей оболочки установлена /sbin/nologin, ничего не происходит. Также в данное поле можно установить значение /bin/false.

В ОС РОСА «КОБАЛЬТ» успешное завершение программы определяется типом возвращаемого значения. Если возвращается нулевое значение, это означает, что выполнение программы прошло успешно. Если ненулевое — значит, в процессе выполнения программы произошли ошибки. На основе возвращаемого значения система аутентификации делает вывод о том, пройдена аутентификация успешно или нет. Программа false независимо от внешних факторов возвращает значение, отличное от нуля, что означает возникновение ошибок при запуске оболочки и возврат управления системе аутентификации.

7. ПРАВА И АТРИБУТЫ ФАЙЛОВ

7.1. Стандартные права и атрибуты

В любой многопользовательской ОС Linux существует понятие прав доступа на файлы, которые подразумевают под собой действия, которые пользователь может совершать, не затрагивая интересы других пользователей. Это набор прав на чтение-запись-выполнение (r-w-x, read-write-execution) для отдельно взятых владельца-группы-других (u-g-o, user-group-other).

Для каждого файла вне зависимости от его типа, будь то обычный файл, каталог, файл псевдо-устройства и т. д., можно посмотреть его атрибуты командой `ls -l`, где в первой колонке в виде последовательности из десяти символов будут указаны права доступа. Первый символ определяет тип файла, а каждые следующие три символа — права доступа, соответственно, для пользователя-группы-остальных.

```
 $ ls -l /var/log/messages
-rw-r--r-- 1 root wheel 44K 14 дек 22:54 /var/log/messages
```

Первый символ «-» означает, что это обычный файл. Другие варианты:

- `d` — каталог;
- `l` — символьная ссылка;
- `c` — символьное устройство;
- `b` — блочное устройство;
- `s` — сокет.

Далее, как видим, для пользователя (root) установлены права на чтение и запись, а для группы (wheel) и остальных — только на чтение. Символы «-» обозначают, что действие, которое должно быть установлено в этой позиции, не разрешено.

Опции «rwx» для файла означают:

- чтение («read») — подразумевается, что можно просматривать содержимое данного файла и копировать файл;
- запись («write») — можно редактировать содержимое файла, но нельзя удалить или переименовать его без соответствующих прав на каталог, в которой находится файл;
- выполнение («execute») — можно запустить файл, если он является исполняемым.

Если рассматривать те же права применительно к каталогу, их смысл меняется:

- r — можно просматривать содержимое каталога;
- w — можно удалять и создавать файлы в данном каталоге;
- x — можно войти в каталог и получить доступ к файлам и подкаталогам для выполнения тех действий, которые для них разрешены (даже если нет права на чтение — достаточно только знать точное название файла).

Кроме символьных обозначений «rwx» существует также числовое представление,

когда установленному «x» соответствует «1», «w» — «2», а «r» — «4», и далее подсчитывается сумма этих трех цифр, соответственно, для владельца, группы и остальных. В вышеописанном случае для файла /var/log/messages получается: для владельца «rw-» → 4+2+0=6, группы и остальных «r--» → 4+0+0=4, т. е. для всего файла будут права «644».

Все возможные варианты прав доступа:

- 0 (000) — ничего не разрешено, ---;
- 1 (001) — запрещено читать и писать, разрешено исполнять, --x;
- 2 (010) — запрещено читать и исполнять, разрешено писать, -w-;
- 3 (011) — запрещено читать, разрешено писать и исполнять, -wx;
- 4 (100) — разрешено читать, запрещено писать и исполнять, r--;
- 5 (101) — разрешено читать и исполнять, запрещено писать, r-x;
- 6 (110) — разрешено читать и писать, запрещено исполнять, rw-;
- 7 (111) — разрешено все, rwx.

Если вы — владелец файла или привилегированный пользователь, вы можете менять права доступа к нему с помощью команды chmod. Можно указывать права либо в числовой форме, либо в формате {кому}{действие}{права}, где {кому} может принимать значения из множества «ugoa» (соответственно: user, group, other, all), {действие} — символ из множества «+-=» (соответственно: добавление, удаление, точная установка прав), а {права} — символ из множества «rwxts» (соответственно: read, write, execute, Sticky-бит, SUID или SGID).

Примеры использования команды chmod для установки/изменения на filename определенных прав:

- дать владельцу права на чтение и запись, группе и остальным — только на чтение:
  `$ chmod 644 filename` или `chmod u=rw,go=r filename`
- добавить для этого файла eXecute для владельца:
  `$ chmod 744 filename` или `chmod u+x filename`
- убрать для группы и всех остальных возможность просматривать этот файл:
  `$ chmod 700 filename` или `chmod go-r filename`

Также можно устанавливать на файлы дополнительные биты:

- Sticky bit;
- SUID;
- SGID.

Для этого необходимо добавить в команде chmod перед трехзначным числом прав дополнительное число (которое по умолчанию равно 0 — «обычный файл») либо, в другом представлении, букву t или s. Если установлен бит SUID, при просмотре атрибутов файла в секции прав доступа для пользователя вместо значения x будет стоять s; если установлен бит SGID — будет поставлен символ s в секции прав для группы; если установлен Sticky bit — в последней позиции секции права доступа для других будет установ-

лен символ `t`.

Sticky bit (бит фиксации) обычно используется только для каталогов и позволяет ограничивать права на запись в них. Если пользователь не является владельцем каталога, но имеет права на запись в него, он может удалять только те файлы в каталоге, владельцем которых он является. Полезно для предотвращения удаления файлов других пользователей в общедоступных каталогах, таких как /tmp. Устанавливается цифрой `1` перед трехзначной комбинацией прав.

SUID (set-user-id — бит смены идентификатора пользователя), установленный на файл, приводит к изменению привилегий запущенного процесса на привилегии владельца исполняемого файла. Исполняемые файлы, владельцем которых является root, с установленным флагом set-user-id запускаются с привилегиями root, даже если их запускает обычный пользователь. Устанавливается цифрой `4` перед трехзначной комбинацией прав. Яркий пример — программа passwd.

SGID (set-group-id — бит смены идентификатора группы), установленный на каталог, приводит к тому, что файлы, создаваемые в этом каталоге, наследуют идентификатор группы каталога, который может не совпадать с идентификатором группы, к которой принадлежит пользователь, создавший файл. Это может быть полезно для каталогов, в которых хранятся файлы, общедоступные для группы пользователей. Устанавливается цифрой `2` перед трехзначной комбинацией прав.

7.2. Специальные файловые атрибуты

Помимо стандартных прав доступа, существуют также дополнительные, или специальные атрибуты файлов, которые поддерживает ФС. Они позволяют осуществлять дополнительный контроль и повышают безопасность системы.

В файловой системе ext2(/3/4) управление специальными атрибутами происходит с помощью команды `chattr`, которая имеет схожий синтаксис с командой изменения прав доступа `chmod`:

```
chattr [-RV] [+-=AacDdijsSu] [-v version] files
```

- `-R` — рекурсивное изменение каталогов и их содержимого;
- `-V` — более подробный вывод;
- `+-=[AacDdijsSu]` — указывает, какие дополнительные атрибуты (биты) должны быть добавлены (`+`), сняты (`-`) или точно установлены (`=`).

7.2.1. Описание атрибутов

- `A` (no atime updates) — не изменять время последнего обращения, что может благоприятно повлиять на производительность ФС, если обращение происходит очень часто;
- `a` (append only) — в файл можно только дописывать (дополнять), но нельзя удалять/переименовывать (удобно для логов). Если этот атрибут установлен на каталог, значит, находящиеся в нем файлы удалять нельзя, но можно создавать новые и модифицировать существующие;

- `c` (compressed) — производится прозрачное сжатие на диске информации файла ядром, а при доступе возвращаются несжатые данные;
- `D` (synchronous directory updates) — при модификации каталога изменения синхронно записываются на диск;
- `d` (no dump) — игнорировать при создании резервной копии программой dump;
- `i` (immutable) — пожалуй, самый используемый и полезный бит, который запрещает любые изменения файла (нельзя удалять, переименовывать и модифицировать файл). Для каталога данный флаг позволяет модифицировать в ней файлы, но запрещает удалять или создавать новые;
- `j` (data journalling) — журналирование данных файла;
- `s` (secure deletion) — полное удаление файла (место на диске, где он находился, после заполняется нулями);
- `S` (synchronous updates) — прямая запись на диск без кэширования (обновление файла на диске производится синхронно с работой приложения, изменяющего данный файл);
- `u` (undeletable) — после удаления файла его содержимое сохраняется, и его можно будет восстановить в дальнейшем.

Для просмотра атрибутов файла используется команда `lsattr`.

Примеры использования:

- установить флаги append-only и immutable. Удобно для файлов, в которые изредка нужно что-либо дописывать, но которые при этом должно быть невозможно удалить. Это достигается снятием и установкой флага immutable в дополнение к уже имеющемуся append-only:
  ```
  chattr +ai <file.txt>
  ```
- снять флаг immutable:
  ```
  chattr -i <file.txt>
  ```
- вывести атрибуты для всех каталогов и файлов:
  ```
  lsattr -a <dir>
  ```
- вывести атрибуты только для каталогов:
  ```
  lsattr -d <dir>
  ```

7.2.2. Управление правами

Управление правами владения и доступа для файлов и каталогов осуществляется с помощью графического приложения «Caja» или с использованием утилит командной строки.

Приложение «Caja» представляет возможность отображения и модификации прав доступа. Полный набор возможностей и описание графического интерфейса приведены в справке приложения. Приложение можно запустить, выбрав пункт меню «Приложения → Системные → Caja» или используя команду `caja`.

7.3. Списки контроля доступа

Последним рассматриваемым дискреционным механизмом ОС являются списки контроля доступа (Access Control List, ACL). Они используются для реализации сложных структур прав доступа и предоставляют больше возможностей, чем стандартный набор полномочий «пользователь-группа-остальные». Возможность использования ACL позволяет администратору получить преимущество от использования более интеллектуальной модели безопасности. По умолчанию ACL не активированы.

Существуют два типа ACL:

1) **ACL для доступа** — список управления доступом для заданного файла или каталога. Проще говоря, это сами права на объект, которые будут контролировать доступ к нему.
2) **ACL по умолчанию** — список управления доступом, связанный только с каталогом. Если файл в этом каталоге не имеет ACL для доступа, он использует правила, определенные в ACL по умолчанию, связанном с каталогом. ACL по умолчанию являются необязательными.

ACL используют расширенные атрибуты для хранения данных о правах доступа к файлам со стороны пользователей и групп. Список управления доступом существует для каждого файла и состоит из шести компонентов. Первые три являются копией стандартных прав доступа к файлу. Они содержатся в единственном экземпляре в ACL и имеются у каждого файла:

- «ACL_USER_OBJ» — режим доступа к файлу пользователя-владельца;
- «ACL_GROUP_OBJ» — режим доступа к файлу группы владельца;
- «ACL_OTHER» — режим доступа к файлу остальных пользователей.

Следующие компоненты устанавливаются для каждого файла в отдельности и могут присутствовать в ACL в нескольких экземплярах:

- «ACL_USER» содержит UID и режим доступа к файлу пользователя, которому установлены права, отличные от остальных. На каждого пользователя со своими правами на данный файл хранится отдельная запись. Не допускается наличие более чем одной записи на одного и того же пользователя;
- «ACL_GROUP» содержит данные тех же типов, что и «ACL_USER», но для группы пользователей;
- «ACL_MASK» содержит маску действующих прав доступа для расширенного режима.

При установке дополнительных прав доступа присваивается значение и элементу «ACL_MASK».

Управление списками ACL осуществляется всего лишь двумя командами:

- `getfacl` — используется для отображения установленных ACL;
- `setfacl` — используется для назначения, модификации и удаления ACL.

7.3.1. Утилита getfacl

Утилита *getfacl* предназначена для отображения ACL файла или каталога.

Синтаксис:

```
getfacl <опции> <файл>
```

В качестве опции может выступать `-R, --recursive` — рекурсивный вывод ACL каталога и его содержимого.

Пример использования:

```
# getfacl qwert
# file: qwert
# owner: root
# group: root
user::rwx
user:child:rw-
group::r--
mask::rw-
other::---
```

Подробное описание утилиты доступно на man-странице getfacl.

7.3.2. Утилита setfacl

Утилита *setfacl* предназначена для назначения, модификации и удаления ACL файлов или каталогов. Синтаксис:

```
setfacl <опции> <ключ> <список_правил> <файл>
```

Часто используемые опции:

- `-b, --remove-all` — удаление списков доступа ACL;
- `-k, --remove-default` — удаление списков доступа ACL по умолчанию;
- `-d, --default` — установка списков доступа ACL по умолчанию;
- `--restore=file` — восстановление списков доступа ACL на объекты из ранее созданного файла с правами;
- `-R, --recursive` — рекурсивное указание списков доступа ACL для каталога и его содержимого.

Поле `<ключ>` обычно задает один из следующих режимов работы:

- `--set` — указание новых списков доступа ACL с удалением всех существующих;
- `-m` — модификация списков доступа ACL;
- `-x` — удаление списков доступа ACL.

Поле `<список_правил>` имеет следующий синтаксис:

- `u:<пользователь>:<права>` — назначение прав для пользователя. Права определяются значениями r, w, x или сочетанием значений;
- `g:<группа>:<права>` — назначение прав для группы (здесь и далее права определяются аналогично);
- `m:<права>` — назначение маски эффективных прав;
- `o:<права>` — назначение прав для прочих пользователей.

Пример использования:

```
# setfacl -m u:allexserv:rwx
qwert
root@sytserver:/media/Work/test# getfacl qwert
# file: qwert
# owner: root
# group: root
user::rwx
user:allexserv:rwx
user:child:rw-
group::r--
mask::rwx
other::---
```

8. ЗАЩИТА SSH-СОЕДИНЕНИЙ

Secure Shell (SSH) — это мощный сетевой протокол, используемый для подключения к другим системам по защищенному каналу. Передача данных по SSH шифруется и защищена от перехвата.

Примечание. В этом разделе рассказывается о наиболее стандартных способах защиты SSH, и предлагаемые здесь способы ни в коем случае не должны считаться исчерпывающими или окончательными. Описание всех значений параметров, доступных для изменения поведения демона sshd, можно просмотреть на странице руководства sshd_config(5), а объяснения базовых принципов работы SSH — на странице руководства ssh(1).

8.1. Криптографический вход в систему

SSH поддерживает использование криптографических ключей для входа в систему. Этот способ гораздо более надежен, чем использование пароля. Сочетание этого способа с другими способами аутентификации может считаться многофакторной аутентификацией.

Использование криптографических ключей для аутентификации возможно, если параметр `PubkeyAuthentication` файла /etc/ssh/sshd_config имеет значение `yes` (так установлено по умолчанию).

Чтобы отключить возможность использования паролей для входа в систему, установите `no` для параметра `PasswordAuthentication`.

Ключи SSH можно создать с помощью команды `ssh-keygen`. При вызове без дополнительных аргументов эта команда создает набор ключей RSA длиной 2048 бит. По умолчанию ключи хранятся в каталоге ~/.ssh/. Для изменения надежности ключа используйте аргумент `-b`. Обычно ключа длиной 2048 бит вполне достаточно.

Теперь в каталоге ~/.ssh/ можно увидеть для ключа. Если при вызове команды `ssh-keygen` были приняты значения по умолчанию, эти два ключа будут называться id_rsa и id_rsa.pub и содержать закрытый и открытый ключи, соответственно. Закрытый ключ необходимо защитить от внешнего воздействия, сделав его нечитаемым для всех, кроме владельца файла. Открытый ключ должен быть перенесен в ту систему, в которую предполагается вход. Для переноса ключа на сервер используйте следующую команду:

```
$ ssh-copy-id -i user@server
```

Эта команда также автоматически добавит открытый ключ в файл ~/.ssh/authorized_keys на сервере. Демон sshd будет проверять этот файл при попытке входа на сервер.

Как и пароли, ключи SSH необходимо регулярно менять. При это не забывайте удалять любые неиспользуемые ключи из файла authorized_keys.

8.2. Методы многофакторной аутентификации

Использование нескольких способов аутентификации, или многофакторная аутентификация, повышает уровень защиты от неавторизованного доступа и поэтому должно

рассматриваться при укреплении системы для защиты от взлома. Для получения доступа к системе, в которой используется многофакторная аутентификация, пользователи должны успешно пройти все этапы аутентификации.

Используемые способы аутентификации указываются в файле /etc/ssh/sshd_config. Обратите внимание, что с помощью его параметров можно указать более одного списка требуемых методов аутентификации, и в таком случае пользователь должен будет успешно пройти каждый метод как минимум из одного списка. Элементы списка разделяются пробелами, а отдельные названия способов аутентификации — запятыми. Например:

```
AuthenticationMethods publickey,gssapi-with-mic
publickey,keyboard-interactive
```

Демон sshd, настроенный с помощью вышеуказанной директивы `AuthenticationMethods`, предоставит доступ только в том случае, если пользователь успешно пройдет аутентификацию по открытому ключу либо в совокупности с gssapi с микрофоном, либо в совокупности с интерактивной аутентификацией с помощью клавиатуры. Обратите внимание, что каждый из запрашиваемых методов аутентификации должен быть явным образом активирован с помощью соответствующей директивы конфигурации (например, `PubkeyAuthentication`) в файле /etc/ssh/sshd_config. Общий список доступных методов аутентификации см. в разделе AUTHENTICATION страницы руководства ssh(1).

8.3. Другие средства защиты SSH

Версия протокола

Хотя реализация протокола SSH, поставляемая в ОС РОСА «КОБАЛЬТ», поддерживает как первую, так и вторую версию протокола для клиентов SSH, только вторая версия должна быть использована везде, где это возможно. Версия SSH-2 содержит многочисленные улучшения по сравнению со старой версией 1, и большинство продвинутых конфигураций возможны только при использовании SSH-2.

Типы ключей

Хотя по умолчанию команда ssh-keygen создает пару ключей SSH-2 RSA, с помощью переданного параметра -t ей можно указать создать также и ключи DSA или ECDSA. Алгоритм ECDSA (Elliptic Curve Digital Signature Algorithm) предоставляет лучшую производительность при той же эквивалентной длине симметричного ключа. Также он создает более короткие ключи.

Порт не по умолчанию

По умолчанию демон sshd слушает TCP порт 22. Смена порта сокращает возможное число уязвимостей системы для атак с использованием автоматического сканирования сети, повышая таким образом защиту по принципу «безопасность через неясность» (security through obscurity). Указать порт можно с помощью директивы `Port` в конфигурационном файле /etc/ssh/sshd_config. Также обратите внимание, что для использования порта не по умолчанию нужно изменять политику SELinux по умолчанию. Это можно сделать, изменив тип SELinux ssh_port_t при помощи следующей команды, выполненной с привилегиями суперпользователя root:

```
# semanage -a -t ssh_port_t -p tcp <номер_порта>
```

Замените <номер_порта> на новый номер, указанный с помощью директивы Port.

Запрет входа в систему под учетной записью root

Если частный случай использования не предусматривает возможности входа в систему под учетной записью root, укажите значение no для директивы PermitRootLogin в файле /etc/ssh/sshd_config. Отключив возможность прямого входа в систему под учетной записью root, системный администратор может проверить, какие именно команды выполняются с полученными привилегиями root, и какие именно пользователи их выполняют, получив доступ в систему и затем получив права root.

Использование PAM для ограничения доступа к службам с привилегиями root

Модуль PAM /lib/security/pam_listfile.so предоставляет очень гибкое средство ограничения доступа для различных учетных записей. Администратор может использовать этот модуль для указания списка пользователей, которым запрещен вход в систему. Для ограничения доступа root к системной службе отредактируйте файл нужной службы в каталоге /etc/pam.d/ так, чтобы для аутентификации требовался модуль pam_listfile.so.

В примере ниже можно увидеть, как этот модуль используется для сервера vsftpd FTP в конфигурационном файле PAM /etc/pam.d/vsftpd (символ \ в конце первой строки необязателен, если директива умещается в одну строку):

```
auth   required  /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Эти параметры указывают PAM обратиться к файлу /etc/vsftpd.ftpusers и отказать в доступе к службе любому из указанных пользователей. Администратор может изменить название этого файла, а также может либо хранить отдельный список для каждой службы, либо использовать один главный список для отказа в доступе ко многим службам.

При необходимости отказать в доступе к нескольким службам аналогичную строку можно добавить в конфигурационные файлы PAM /etc/pam.d/pop и /etc/pam.d/imap (для почтовых клиентов) или /etc/pam.d/ssh (для клиентов SSH).

9. УСТАНОВКА ОГРАНИЧЕНИЙ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ

Встроенная в оболочку утилита *ulimit* позволяет устанавливать и отображать ограничения для учетных записей пользователей. Перечень доступных пользователю системных ресурсов задается в файле /etc/security/limits.conf. Он имеет следующий вид:

```
#<domain> <type> <item> <value>
*     - core <value>
*     - data <value>
*     - priority <value>
*     - fsize <value>
*     soft sigpending <value> eg:57344
*     hard sigpending <value> eg:57444
*     - memlock <value>
*     - nofile <value> eg:1024
*     - msgqueue <value> eg:819200
*     - locks <value>
*     soft core <value>
*     hard nofile <value>
@<group> hard nproc <value>
<user>  soft nproc <value>
%<group> hard nproc <value>
<user>  hard nproc <value>
@<group> - maxlogins <value>
<user>  hard cpu <value>
<user>  soft cpu <value>
<user>  hard locks <value>
```

<domain> может быть:

- именем пользователя;
- именем группы с синтаксисом `@group`;
- символом подстановки `*`, обозначающим значение по умолчанию;
- символом подстановки `%`. Также он может использоваться с синтаксисом `%group` для указания максимального ограничения входов в систему.

<type> может иметь два значения:

- `soft` — для принудительного применения «мягких» ограничений. Мягкое ограничение определяет число системных ресурсов, которое пользователь еще может превысить;
- `hard` — для принудительного применения «жестких» ограничений. Жесткое ограничение превысить невозможно: при попытке сделать это пользователь получит сообщение об ошибке.

<item> может иметь одно из следующих значений:

- `core` — ограничивает размер файлов дампа памяти (КБ);
- `data` — максимальный размер данных (КБ);

- `fsize` — максимальный размер файла (КБ);
- `memlock` — максимальный размер адресного пространства, закрепленного в памяти (КБ);
- `nofile` — максимальное число открытых файлов;
- `rss` — максимальный размер страниц памяти (КБ);
- `stack` — максимальный размер стека (КБ);
- `cpu` — максимальное время ЦП (минут);
- `nproc` — максимальное число процессов;
- `as` — предел адресного пространства (КБ);

- `maxlogins` — максимальное число одновременных регистраций пользователя в системе;
- `maxsyslogins` — максимальное число входов в эту систему;
- `priority` — приоритет, с которым выполняется процесс пользователя;
- `locks` — максимальное число блокировок файлов на этого пользователя;
- `sigpending` — максимальное число ожидающих сигналов;
- `msgqueue` — максимальный объем памяти, используемый очередью сообщений POSIX (байтов);
- `nice` — максимальный разрешенный приоритет в пределах значений [−20, 19];
- `rtprio` — максимальный приоритет реального времени.

Для применения изменений выйдите из терминала и снова выполните вход. Подробности см. на man-странице limits.conf.

Пример: ограничение числа параллельных сеансов доступа

```
* - maxlogins 5
```

При попытке регистрации в шестом сеансе пользователь user увидит следующее сообщение:

```
Too many logins for 'user'.
```

Пример: установка приоритета процессов по умолчанию

```
* hard priority -2
```

Пример: ограничение приоритета процессов

- `hard nice -4`

9.1. Квоты дискового пространства

Для выделения пользователю строго определенного дискового пространства (квоты) необходимо включить поддержку квот в ФС. Для этого нужно отредактировать файл /etc/fstab: добавить опцию `usrquota` и/или `grpquota` при монтировании ФС, в которых требуется включить поддержку квот. Параметр `usrquota` обозначит поддержку квот для пользователей, а `grpquota` — для групп.

9.1.1. Конфигурационный файл /etc/fstab

Файл /etc/fstab содержит информацию обо всех разделах жесткого диска и других

носителях информации, установленных в системе. Формат файла:

```
<file system>  <dir>  <type>  <options>  <dump>  <pass>
```

Ниже кратко описаны основные поля файла /etc/fstab. Чтобы получить подробное описание, выполните команду `man fstab`.

1) `<file system>` — ФС (имя файла устройства, идентификатор устройства UUID или метка тома LABEL).
2) `<dir>` — точка монтирования системы.
3) `<type>` — тип ФС (ext3, ext4, xfs и др.).
4) `<options>` — опции монтирования:
   - `defaults` (`rw`, `suid`, `dev`, `exec`, `auto`, `nouser` и `async`);
   - `auto` — монтирование ФС при загрузке происходит автоматически или после выполнения команды `mount -a`;
   - `noauto` — монтирование ФС разрешено только вручную;
   - `async` — все операции ввода/вывода должны выполняться асинхронно;
   - `exec` — исполнение бинарных файлов разрешено;
   - `noexec` — исполнение бинарных файлов запрещено;
   - `user` — монтирование ФС разрешено любому пользователю (применяются опции `noexec`, `nosuid`, `nodev`, если они не переопределены);
   - `nouser` — монтирование ФС разрешено только суперпользователю (используется по умолчанию);
   - `suid` — операции с битами suid и sgid разрешены;
   - `nosuid` — операции с битами suid и sgid запрещены;
   - `usrquota` — поддержка квот для пользователей;
   - `grpquota` — поддержка квот для групп.
5) `<dump>` — флаг резервного копирования (0 — выполнять, 1 — не выполнять).
6) `<pass>` — порядок проверки ФС утилитой *fsck* (0 — нет проверки, 1 — высокий приоритет, 2 — низкий приоритет. Обычно приоритет 1 присваивают корневой ФС.

После редактирования файла /etc/fstab нужно перезагрузить систему или перемонтировать ФС, записи которых были изменены, выполнив следующую команду с привилегиями суперпользователя root:

```
# mount -o remount <файловая_система>
```

В качестве аргумента `<файловая_система>` указывается ФС, запись которой была изменена, или точка монтирования ФС.

9.1.2. Проверка квот

После того, как все ФС, в которых включены поддержки квот, перемонтированы, ОС может работать с дисковыми квотами. Следующим действием должен быть запуск утилиты *quotacheck*. Утилита quotacheck предназначена для проверки ФС, в которых включена поддержка квот, и обновления таблицы текущего использования диска в ФС. Затем эта таблица используется для обновления системной копии данных об использовании диска.

Синтаксис:

```
quotacheck <опции> <файловая_система>
```

Ниже кратко описаны часто используемые опции утилиты *quotacheck*. Чтобы получить подробное описание, выполните команду `man quotacheck`.

- `-c`, `--create-files` — создание файлов квот;
- `-a`, `--all` — проверка всех локально смонтированных ФС, в которых включена поддержка квот;
- `-u`, `--user` — поддержка дисковых квот пользователей;
- `-g`, `--group` — поддержка дисковых квот групп;
- `-v`, `--verbose` — вывод подробной информации о процессе проверки квот.

Чтобы создать в ФС файлы квот и проверить их, выполните:

```
# quotacheck -cug <file system>
# quotacheck -avug
```

9.1.3. Выделение квот

Для выделения дисковых квот предназначена утилита *edquota*.

Синтаксис:

```
edquota <опции>
```

Ниже кратко описаны часто используемые опции утилиты *edquota*. Чтобы получить подробное описание, выполните команду `man edquota`.

- `-u`, `--user` — поддержка дисковых квот пользователей;
- `-g`, `--group` — поддержка дисковых квот групп;
- `-t`, `--edit-period` — редактирование периода отсрочки;
- `-T`, `--edit-times` — редактирование периода отсрочки для пользователя/группы.

Чтобы настроить квоты для пользователя user, выполните:

```
# edquota -u user
```

Команда выводит квоту пользователя и предоставляет возможность отредактировать выделенные квоты в текстовом редакторе:

```
Disk quotas for user user (uid 1000):
Filesystem     blocks     soft     hard    inodes   soft   hard
  /dev/sdb      29824        0     2000         1      0      0
```

В первом столбце приводится название ФС, для которой включена поддержка квот. Во втором столбце указывается, сколько блоков использует пользователь в данный момент, в следующих двух столбцах — мягкое и жесткое ограничения на число блоков для пользователя в данной ФС. В столбце `inodes` указано, сколько дескрипторов inodes использует пользователь, в следующих двух столбцах — мягкое и жесткое ограничения на число inode для пользователя в данной ФС.

Жесткий предел определяет абсолютный максимальный объем дискового пространства, которое может быть выделено пользователю. Если этот предел достигнут, получить дополнительное дисковое пространство невозможно. Мягкий предел также определяет максимальный объем дискового пространства. Однако, в отличие от жесткого преде-

ла, мягкий предел может быть превышен в течение некоторого времени. Это время называется периодом отсрочки. Период отсрочки можно задавать в секундах, минутах, часах, днях, неделях или месяцах. Если одно из этих значений равно 0, предел не устанавливается.

9.1.4. Просмотр квот

Для просмотра квот предназначена утилита *quota*.

Синтаксис:

```
quota <опции>
```

Ниже кратко описаны часто используемые опции утилиты *quota*. Чтобы получить подробное описание, выполните команду `man quota`.

- `-u`, `--user` — дисковые квоты пользователя;
- `-g`, `--group` — дисковые квоты группы.

Чтобы проверить, установились ли квоты для пользователя, выполните:

```
$ quota -u user
```

9.1.5. Включение и отключение поддержки квот

Для включения и отключения поддержки квот предназначены утилиты *quotaon* и *quotaoff*.

Синтаксис:

```
quotaon <опции>
quotaoff <опции>
```

Ниже кратко описаны часто используемые опции утилит *quotaon* и *quotaoff* (их списки не отличаются). Подробные описания приведены на man-страницах quotaon и quotaoff.

- `-u`, `--user` — дисковые квоты пользователя;
- `-g`, `--group` — дисковые квоты группы;
- `-a`, `--all` — включение/отключение дисковых квот на всех ФС.

Вышеописанные утилиты используются при выделении квот на ФС типа ext3 и ext4. При выделении квот на ФС типа xfs утилиты `quotacheck`, `quotaon` и `quotaoff` не используются.

10. НАСТРОЙКА ВРЕМЕНИ И ДАТЫ

В современных операционных системах различают два типа часов.

1) Часы реального времени (Real Time Clock, RTC), которые обычно называются «аппаратными часами» и чаще всего представляют собой интегральную схему на материнской плате. Аппаратные часы совершенно независимы от текущего состояния ОС и продолжают работу даже в выключенном компьютере.
2) Системные часы, также известные как «программные часы», работа которых поддерживается ядром и изначальное значение которых основано на показаниях аппаратных часов. После загрузки системы и инициализации системных часов они работают совершенно независимо от аппаратных.

Системное время всегда хранится в формате UTC («всемирное координированное время») и по необходимости преобразуется в местное время в приложениях. Местное время — это время часового пояса, установленного в системе, с учетом перехода на летнее время (DST, daylight saving time). Аппаратные часы могут быть установлены либо на UTC, либо на местное время. Рекомендуется использовать UTC.

В составе ОС РОСА «КОБАЛЬТ» есть три консольные утилиты, которые можно использовать для настройки и отображения информации системного времени и даты: новая утилита *timedatectl*, являющаяся частью systemd, традиционная утилита *date* и утилита *hwclock*, предназначенная для доступа к аппаратным часам.

10.1. Использование утилиты *timedatectl*

Утилита *timedatectl* поставляется в составе systemd и позволяет просматривать и изменять параметры системных часов. Эту утилиту можно использовать для изменения текущего времени и даты, для установки часового пояса или для включения автоматической синхронизации системных часов с удаленным сервером.

Сведения о том, как отобразить текущее время и дату в пользовательском формате, см. также в подразделе 10.2. «Использование утилиты date».

10.1.1. Просмотр текущего времени и даты

Чтобы просмотреть текущее время и дату, а также дополнительную информацию о параметрах системных и аппаратных часов, выполните команду `timedatectl` без параметров.

Будет показано местное и универсальное время, текущий используемый часовой пояс, статус параметров протокола сетевого времени (NTP) и дополнительная информация о переходе на летнее время.

Пример: просмотр текущего времени и даты

Ниже показан пример вывода команды `timedatectl` в системе, не использующей NTP для синхронизации системных часов с удаленным сервером:

```
$ timedatectl
      Local time: Mon 2016-09-16 19:30:24 CEST
  Universal time: Mon 2016-09-16 17:30:24 UTC
```

```
        Timezone: Europe/Prague (CEST, +0200)
     NTP enabled: no
NTP synchronized: no
 RTC in local TZ: no
      DST active: yes
 Last DST change: DST began at
                  Sun 2016-03-31 01:59:59 CET
                  Sun 2016-03-31 03:00:00 CEST
 Next DST change: DST ends (the clock jumps one hour backwards)
at
                  Sun 2016-10-27 02:59:59 CEST
                  Sun 2016-10-27 02:00:00 CET
```

Примечание. `timedatectl` не сразу замечает изменения в статусе chrony или ntpd. Если были внесены изменения в параметры или в статус этих инструментов, выполните следующую команду:

```
# systemctl restart systemd-timedated.service
```

10.1.2. Изменение текущего времени

Чтобы изменить текущее время, выполните следующую команду с привилегиями суперпользователя root:

```
timedatectl set-time ЧЧ:ММ:СС
```

Замените ЧЧ на часы, ММ — на минуты и СС — на секунды, все в двузначном формате.

Эта команда обновляет как системные, так и аппаратные часы. Результат аналогичен результатам двух команд: `date --set` и `hwclock --systohc`.

При активной службе NTP команда не выдаст успешного результата. Чтобы временно отключить службу NTP, см. подраздел 10.4. «Синхронизация системных часов с удаленным сервером».

Пример: смена текущего времени

Чтобы сменить текущее время на 11 часов 26 минут пополудни, выполните следующую команду с привилегиями суперпользователя root:

```
# timedatectl set-time 23:26:00
```

По умолчанию в системе используется время UTC. Чтобы настроить систему на поддержание местного времени, выполните:

```
# timedatectl set-local-rtc <boolean>
```

Вместо `<boolean>` укажите `yes`, `y`, `true`, `t` или `1`.

10.1.3. Смена текущей даты

Чтобы изменить текущую дату, выполните следующую команду с привилегиями суперпользователя root:

```
# timedatectl set-time ГГГГ-ММ-ДД
```

Замените `ГГГГ` на четырехзначное значение года, `М` — на двузначное значение ме-

сяца, а `ДД` — на двузначное значение дня месяца. Обратите внимание, что смена даты без указания текущего времени установит значение 00:00:00.

Пример: смена текущей даты

Чтобы сменить текущую дату на 2 июня 2017 года и сохранить текущее время (11 часов 26 минут пополудни), выполните следующую команду с привилегиями суперпользователя root:

```
# timedatectl set-time 2017-06-02 23:26:00
```

10.1.4. Смена часового пояса

Чтобы получить список всех доступных часовых поясов, выполните следующую команду с привилегиями суперпользователя root:

```
# timedatectl list-timezones
```

Чтобы изменить используемый часовой пояс, выполните:

```
# timedatectl set-timezone <часовой_пояс>
```

Замените `<часовой_пояс>` на любое значение, выведенное в результате выполнения команды `list-timezones`.

Пример: смена часового пояса

Чтобы определить, какой часовой пояс наиболее близко расположен к вашему текущему местоположению, используйте команду `timedatectl` с ключом `list-timezones`. Чтобы, например, поучить список всех доступных часовых поясов Европы, выполните:

```
# timedatectl list-timezones | grep Europe
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
...
```

Чтобы сменить часовой пояс на Europe/Prague (Европа, г. Прага), выполните:

```
# timedatectl set-timezone Europe/Prague
```

10.2. Использование утилиты *date*

Утилита *date* доступна во всех системах Linux и дает возможность просмотреть и настроить текущие время и дату. Часто ее используют в сценариях для показа подробной информации о системных часах в пользовательском формате.

Информацию о том, как сменить часовой пояс или включить автоматическую синхронизацию системных часов с удаленным сервером, см. в подразделе 10.1. «Использование утилиты timedatectl».

10.2.1. Просмотр текущей даты и времени

Чтобы просмотреть текущие дату и время, выполните команду `date` без дополни-

тельных аргументов. Будет оказан день недели, затем текущая дата, местное время, краткое название часового пояса, а также год.

По умолчанию команда `date` показывает местное время. Чтобы посмотреть время в формате UTC, запустите команду с ключом `--utc` или `-u`. Также можно настроить формат показываемой информации, указав параметр `+"format"`:

```
# date +"format"
```

Замените `format` одной или несколькими управляющими последовательностями. Список наиболее часто употребляемых параметров форматирования см. ниже.

Наиболее часто употребляемые управляющие последовательности

- %H — час в формате ЧЧ (например, 17);
- %M— минута в формате ММ (например, 30);
- %S — секунда в формате СС (например, 24);
- %d — день месяца в формате ДД (например, 16);
- %m— месяц в формате ММ (например, 09);
- %Y — год в формате ГГГГ (например, 2018);
- %Z — краткое обозначение часового пояса (например, CEST);
- %F — дата полностью в формате ГГГГ-ММ-ДД (например, 2018-09-16). Этот параметр аналогичен параметру `%Y-%m-%d`;
- %T — время полностью в формате ЧЧ:ММ:СС (например, 17:30:24). Этот параметр аналогичен параметру `%H:%M:%S`.

Пример: показ текущей даты и времени

Чтобы просмотреть текущую дату и время, выполните следующую команду:

```
$ date
Mon Sep 16 17:30:24 CEST 2018
```

Чтобы просмотреть текущую дату и время в формате UTC, выполните:

```
$ date --utc
Mon Sep 16 15:30:34 UTC 2018
```

Чтобы получить пользовательский формат вывода данных команды `date`, выполните:

```
$ date +"%Y-%m-%d %H:%M"
2018-09-16 17:30
```

10.2.2. Смена текущего времени

Чтобы сменить текущее время, выполните следующую команду с привилегиями суперпользователя root:

```
# date --set "ЧЧ:ММ:СС"
```

Замените `ЧЧ` на часы, `ММ` — на минуты, а `СС` — на секунды, все в двузначном формате.

По умолчанию команда `date` устанавливает системные часы на местное время. Чтобы установить время в формате UTC, добавьте параметр `--utc` или `-u`:

```
# date --set "ЧЧ:ММ:СС" --utc
```

10.2.3. Смена текущей даты

Чтобы сменить текущую дату, выполните следующую команду с привилегиями суперпользователя root:

```
# date --set "ГГГГ-ММ-ДД"
```

Замените `ГГГГ` на год (четыре цифры), `ММ` — на порядковый номер месяца, а `ДД` — на порядковый номер дня месяца. Обратите внимание, что смена дат без указания текущего времени установит время на 00:00:00.

Пример: смена текущей даты

Чтобы сменить текущую дату на 2 июня 2018 г. и оставить текущее время (23 часа 26 минут), выполните:

```
# date --set "2018-06-02 23:26:00"
```

10.3. Использование утилиты hwclock

hwclock — это утилита для доступа к аппаратным часам, также называемым часами реального времени (RTC). Аппаратные часы не зависят от используемой ОС и продолжают работать даже после выключения компьютера (от батарейки на материнской плате). Утилита *hwclock* используется для показа времени аппаратных часов. *hwclock* также может компенсировать систематическое смещение аппаратных часов.

Аппаратные часы хранят следующие значения: год, месяц, день, час, минута и секунда. Аппаратные часы не могут хранить стандарт времени, местное время или время UTC, а также не могут устанавливать летнее время.

Утилита *hwclock* хранит свои параметры в файле /etc/adjtime, который создается при первом внесении изменений, например, при ручном изменении времени или при синхронизации аппаратных часов с системным временем.

Примечание. В ОС РОСА «КОБАЛЬТ», если системные часы синхронизируются с помощью протокола сетевого времени NTP или протокола точного времени PTP, ядро выполняет автоматическую синхронизацию аппаратных часов и системных часов каждые 11 минут.

10.3.1. Просмотр текущего времени и даты аппаратных часов

Выполнение `hwclock` без параметров с привилегиями суперпользователя root возвращает дату и время в формате местного времени.

Обратите внимание, что использование ключа --utc или -u с командой hwclock не означает, что аппаратное время будет показано в формате UTC или местного времени. Эти ключи используются для хранения/отсчета времени в одном из двух форматов. hwclock всегда показывает местное время. Кроме того, использование команд hwclock --utc или hwclock --local не изменяет запись в файле /etc/adjtime. Эта команда может пригодиться, когда известно, что значение, сохраненное в /etc/adjtime, не является правиль-

ным, но его не нужно изменять. С другой стороны, в случае некорректной формы команды можно получить неверную информацию.

Пример: просмотр даты и времени аппаратных часов

Чтобы просмотреть текущую дату и текущее местное время аппаратных часов, выполните:

```
# hwclock
Tue 15 Apr 2017 04:23:46 PM CEST     -0.329272 seconds
```

CEST — это краткое обозначение часового пояса, означающее Central European Summer Time (центрально-европейское летнее время).

Информацию о том, как изменить часовой пояс, см. в п. 10.1.4. «Смена часового пояса».

10.3.2. Настройка даты и времени

Помимо просмотра даты и времени, аппаратные часы позволяют вручную установить конкретное время.

При необходимости сменить время и дату аппаратных часов это можно сделать при помощи параметров `--set` и `--date` и конкретного значения:

```
# hwclock --set --date "ДД МММ ГГГГ ЧЧ:ММ"
```

Замените `ДД` на день (двузначное число), `МММ` — на месяц (трехбуквенное сокращение), `ГГГГ` — на год (четырехзначное число), `ЧЧ` — на час (двузначное число), а `ММ` — на минуту (двузначное число).

В то же время можно настроить аппаратные часы на отсчет времени либо в значении UTC, либо в значении местного времени. Это выполняется при помощи ключей `--utc` и `--localtime`, соответственно. В этом случае в файл `/etc/adjtime` записывается UTC или LOCAL.

Пример: настройка аппаратных часов на конкретные дату и время

При необходимости настроить дату и время на конкретное значение, например, 21 октября 2018 г., 21:17, и сохранить формат аппаратных часов UTC, выполните:

```
# hwclock --set --date "21 Oct 2018 21:17" --utc
```

10.3.3. Синхронизация даты и времени аппаратных часов и времени ОС

Синхронизацию аппаратных часов и текущего системного времени можно выполнять в двух направлениях.

Во-первых, можно настроить значение аппаратных часов на текущее системное время при помощи следующей команды:

```
# hwclock --systohc
```

Обратите внимание, что в случае использования NTP аппаратные часы синхронизируются с с системными часами каждые 11 минут, и эта команда имеет смысл только при загрузке (для получения приемлемых значений начального системного времени).

Во-вторых, можно установить системное время на значение аппаратных часов с помощью следующей команды:

```
# hwclock --hctosys
```

При синхронизации аппаратных часов с системным временем также можно указать, должны ли аппаратные часы придерживаться местного времени или UTC, добавив ключ `--utc` или `--localtime`. По аналогии с использованием ключа `--set` в файл /etc/adjtime будет записано значение UTC или LOCAL.

Результат выполнения команды `hwclock --systohc --utc` аналогичен результату команды `timedatectl set-local-rtc false`, а команда `hwclock --systohc --local` является альтернативой команде `timedatectl set-local-rtc true`.

10.3.4. Синхронизация аппаратных часов с системным временем

Чтобы настроить аппаратные часы на текущее системное время и сохранить локальное время в аппаратных часах, выполните следующую команду с привилегиями суперпользователя root:

```
# hwclock --systohc --localtime
```

Чтобы избежать возможных проблем с часовым поясом и переходом на летнее время, рекомендуется настраивать аппаратные часы на формат UTC. Пример «Синхронизация аппаратных часов с системным временем» удобен, например, в системах с несколькими ОС, одна из которых — ОС Windows, которая по умолчанию предполагает работу аппаратных часов со значением местного времени, а все другие ОС должны подстраиваться под это поведение, также используя местное время. Также это может пригодиться при работе с виртуальными машинами: если виртуальные аппаратные часы, предоставляемые хостом, показывают местное время, гостевую ОС также нужно настроить на использование местного времени.

10.4. Синхронизация системных часов с удаленным сервером

Выше была рассмотрена ручная настройка времени. Теперь опишем процесс настройки автоматической синхронизации с NTP-сервером.

Команда `timedatectl` также дает возможность включить автоматическую синхронизацию системных часов с группой удаленных серверов, используя протокол NTP. Активация NTP запускает службу chronyd или службу ntpd в зависимости от того, какая из них установлена в системе.

Включить или отключить службу NTP можно с помощью следующей команды:

```
# timedatectl set-ntp <boolean>
```

Чтобы включить синхронизацию системных часов с удаленным сервером NTP, замените `<boolean>` на `yes` (это значение по умолчанию), а чтобы отключить — на `no`.

По умолчанию в ОС РОСА «КОБАЛЬТ» используется служба времени chronyd, которая уже установлена и запускается по умолчанию.

10.4.1. Краткое описание Chrony

Chrony — альтернативный клиент и сервер NTP, адаптированный для задач роуминга и разработанный специально для систем без постоянного присутствия в сети.

Что нужно знать, выбирая между Chrony и NTP:

- в некоторых современных ОС, таких, как ОС РОСА «КОБАЛЬТ», chronyd является демоном NTP по умолчанию, заменив ntpd;
- ntpd по-прежнему находится в репозиториях yum для систем, где нужно использовать службу NTP;
- Chrony является реализацией протокола сетевого времени NTP, отличной от демона ntpd. Среди его особенностей — возможность более быстрой синхронизации времени, чем у ntpd, а также повышенная точность этой синхронизации.

Преимущества Chrony

1) Ускоренная синхронизация. Для минимизации ошибок времени и погрешностей частоты требуются минуты, а не часы, что удобно для рабочих станций или часто отключаемых машин, не работающих 24 часа в сутки.
2) Повышенная отзывчивость к появлениям погрешностей частоты, что удобно для виртуальных машин с нестабильными часами или для технологий сбережения энергии без постоянной тактовой частоты.
3) Синхронизация выполняется только один раз в начале работы службы, во избежания отрицательного влияния на те приложения, которым нужно монотонное время.
4) Повышенная стабильность при работе с асимметричными задержками времени, например, при долгих загрузках в перегруженной сети.
5) Регулярные опросы серверов не требуются, поэтому в системах без постоянного сетевого подключения синхронизация выполняется быстро.

Когда лучше использовать chrony

Chrony считается лучшим выбором для систем, которые часто приостанавливают работу или временно отключаются от сети (серверы мобильной связи, виртуальные серверы и так далее).

Когда лучше использовать NTP

Демон NTP (ntpd) лучше использовать в системах, которые работают постоянно и без отключения. В системах, где необходимо использовать широковещательный или групповой адрес IP, или где аутентификация пакетов выполняется по автоключу, также желательно использовать ntpd.

В конфигурационном файле данной службы `/etc/chronyd.conf` указаны два сервера, с которых происходит синхронизация:

- ntp.rosalinux.ru;
- ntp2.rosalinux.ru.

Текущий список серверов, используемый данной службой, можно получить следующей командой:

```
# chronyc sources
```

В качестве вывода данной команды вы должны получить список серверов точного времени:

```
210 Number of sources = 1
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================
```

```
========
^* ntp.rosalinux.ru              2   7   377   201  -5050ns[
-72us] +/-   66ms
```

Узнать статус синхронизации времени можно следующей командой:

```
# chronyc tracking
```

В качестве вывода данной команды вы должны получить ряд сведений о работе сервиса:

```
Reference ID    : 195.19.76.82 (ntp.rosalinux.ru)
Stratum         : 3
Ref time (UTC)  : Tue Oct 24 11:08:21 2017
System time     : 0.000004665 seconds slow of NTP time
Last offset     : -0.000004627 seconds
RMS offset      : 0.000251306 seconds
Frequency       : 0.394 ppm fast
Residual freq   : +0.003 ppm
Skew            : 0.916 ppm
Root delay      : 0.094890 seconds
Root dispersion : 0.014990 seconds
Update interval : 65.0 seconds
Leap status     : Normal
```

В данном примере время в формате UTC было успешно синхронизировано с сервером ntp.rosalinux.ru. Если в поле Stratum отображается 16, значит, синхронизация с источником работает неправильно.

10.4.2. Использование клиента NTP

Если вместо chrony вы хотите использовать классический NTP, нужно выполнить следующие шаги:

1) Установить сервис ntp, если он еще не установлен:

   ```
   # yum install ntp
   ```

2) Отключить chrony. Чтобы использовать ntpd в качестве службы сетевого времени по умолчанию, необходимо остановить и отключить демон chronyd. Выполните следующую команду с привилегиями суперпользователя root:

   ```
   # systemctl stop chronyd
   ```

   Чтобы предотвратить запуск chronyd во время загрузки системы, выполните следующую команду с привилегиями суперпользователя root:

   ```
   # systemctl disable chronyd
   ```

3) Включить службу NTP:

   ```
   # systemctl start ntpd
   ```

   Чтобы добавить NTP в автозагрузку, выполните:

   ```
   # systemctl enable ntpd
   ```

Если ваш сервер не подключен к интернету и/или вы планируете использовать свои сервера точного времени, внесите изменения в конфигурационный файл NTP

(/etc/ntp.conf), указав в параметре `server` IP-адрес или полное имя (FQDN) вашего сервера. По умолчанию в нем содержатся только сервера ntp.rosalinux.ru и ntp2.rosalinux.ru. После чего следует перезапустить службу NTP:

```
# systemctl restart ntpd
```

Чтобы при загрузке ОС ваш клиент обновлял время с сервера NTP, добавьте записи о них в файл /etc/ntp/step-tickers.

4) Проверить работу сервера точного времени ntpd. Проверка возможна только с клиентского АРМ, для этого используется следующая команда:

```
# ntpq -p
```

Она должна вывести сведения об используемом вами сервере NTP:

```
     remote           refid      st t when poll reach   delay   offs    jitter
==============================================================================
*ntp.rosalinux.ru  .LOCL.          1 u   22   64  377    0.209   26.957   1.973
```

Доступная локально документация по настройке даты и времени

- timedatectl(1) — страница руководства по консольной утилите *timedatectl*. Содержит информацию о том, как выполнять запросы и настраивать параметры системных часов;
- date(1) — страница руководства команды `date`. Предоставляет полный список поддерживаемых параметров командной строки;
- hwclock(8) — страница руководства команды `hwclock`. Предоставляет полный список поддерживаемых параметров командной строки.

11. АУДИТ

Система аудита в ОС РОСА «КОБАЛЬТ» представляет собой средство отслеживания информации, имеющей отношение к безопасности системы. Основанная на предварительно настроенных правилах, программа Audit создает записи в журнале с целью зафиксировать такое количество информации о событиях, происходящих в системе, какое только возможно. Без этой информации, как правило, невозможно определить нарушителя политики безопасности в критически важных окружениях.

Audit не предоставляет дополнительные средства безопасности системы, но ее можно использовать для раскрытия нарушений политик безопасности, уже используемых в системе. Повторение замеченных с помощью Audit нарушений в дальнейшем может быть предотвращено с использованием дополнительных мер безопасности.

В списке ниже приведены примеры информации, которую может записать Audit в файлы своего журнала:

- дата, время, тип и результат события;
- метки конфиденциальности субъектов и объектов;
- связь события с идентификатором пользователя, вызвавшего событие;
- все изменения файлов Audit и попытки поучения доступа к файлам журнала Audit;
- все случаи использования механизмов аутентификации, таких как SSH, Kerberos и др.;
- изменения в любой доверенной базе данных, например, в файле /etc/passwd;
- попытки импортировать или экспортировать информацию в/из системы;
- включение или исключение событий на основе идентификатора пользователя, меток субъектов и объектов и других атрибутов.

11.1. Варианты использования Audit

Отслеживание доступа к файлам

Audit может отследить, был ли выполнен доступ к файлу или к каталогу, вносились ли изменения, запускался ли файл, были ли изменены атрибуты файла. Это удобно, например, для обнаружения доступа к важным файлам. В случае повреждения таких файлов администратор обращается к информации, записанной Audit.

Наблюдение за системными вызовами

Audit можно настроить на создание записи в журнале каждый раз, когда используется конкретный системный вызов. Эту возможность можно использовать, например, для отслеживания изменения в системном времени, выполняя наблюдения за settimeofday, clock_adjtime и другими системными вызовами, имеющими отношение к времени.

Запись команд, выполняемых пользователем

Audit отслеживает запуск файлов на выполнение, поэтому администраторы имеют возможность настроить правила на запись каждого выполнения конкретной команды. Правило можно создать, например, для каждого выполняемого файла в каталоге /bin. По полученным записям в журнале далее можно выполнить поиск по идентификатору пользо-

вателя и создать журнал аудита, содержащий выполненные пользователем команды.

Запись событий безопасности

Модуль аутентификации pam_faillock может записывать неудачные попытки входа в систему. Audit также можно настроить на запись неудачных попыток входа в систему. В этом случае Audit будет предоставлять дополнительную информацию о пользователе, который сделал попытку входа.

Поиск событий

Audit предоставляет утилиту ausearch, которую можно использовать для фильтрации записей в журнале и получения полного журнала аудита на основе некоторого количества условий.

Получение сводных отчетов

Утилиту aureport, помимо других возможностей, можно использовать для создания ежедневных отчетов о записанных событиях. Системный администратор затем может проанализировать эти отчеты и далее глубже исследовать подозрительную активность.

Наблюдение за сетевым доступом

Утилиты iptables и ebtables можно настроить на запуск событий аудита, что дает администратору возможность наблюдения за сетевым доступом.

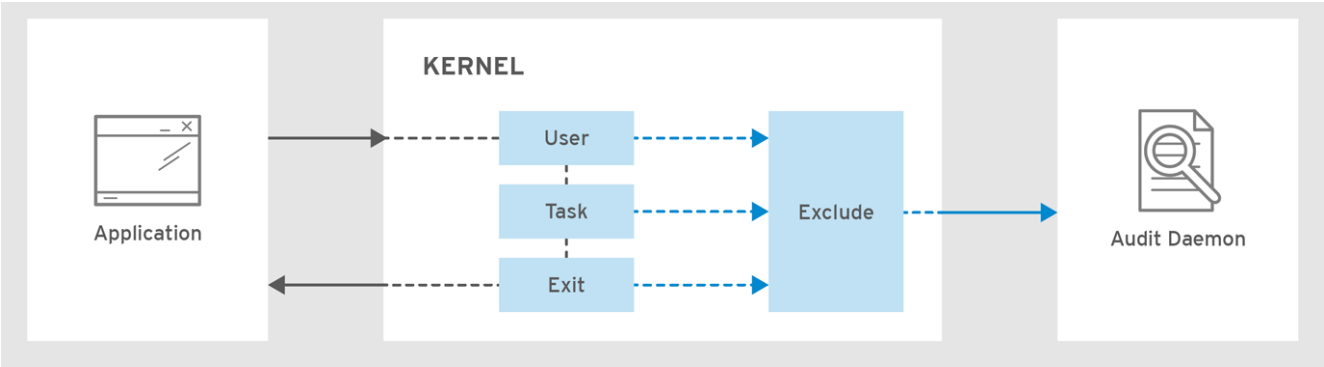
Примечание. В зависимости от объема информации, собираемого программой Audit, она может оказывать влияние на производительность системы.

11.2. Архитектура системы Audit

Программа Audit включает в себя:

1) Приложения и утилиты в пространстве пользователя.
2) Обработчик системных вызовов со стороны ядра.

Компонент ядра получает системные вызовы от приложений из пространства пользователя и пропускает их сквозь один из трех фильтров: user, task или exit. Как только системный вызов проходит фильтр исключения, он посылается сквозь один из вышеуказанных фильтров, который, основываясь на параметрах правила Audit, передает его демону аудита для дальнейшей обработки. Рисунок ниже иллюстрирует этот процесс.



Архитектура системы аудита

Демон Audit, расположенный в пространстве пользователя, собирает информацию ядра и создает записи в файле журнала. Другие утилиты Audit в пользовательском про-

странстве взаимодействуют с демоном Audit, с компонентом ядра Audit или с файлами журнала Audit:

audisp — демон-координатор программы Audit, который взаимодействует с демоном Audit и посылает события другим приложениям для дальнейшей обработки. Назначение этого демона — предоставление механизма интеграции, так чтобы аналитические программы реального времени могли взаимодействовать с событиями Audit.

auditctl — утилита контроля Audit, которая взаимодействует с компонентом ядра для управления правилами и для контроля некоторого числа установок и параметров процесса создания событий.

Остальные утилиты Audit получают содержимое журнала Audit в качество ввода и создают вывод на основе требований пользователя. Утилита aureport, например, создает отчет обо всех записанных событиях.

11.3. Установка пакетов Audit

В ОС РОСА «КОБАЛЬТ» пакет аудита устанавливается по умолчанию. Но если вы его удалили, его можно установить заново с DVD следующей командой:

```
# yum install audit
```

Установка параметров Audit

Демон Audit настраивается в файле /etc/audit/auditd.conf. Этот файл состоит из конфигурационных параметров, изменяющих поведение демона Audit. Пустые строки и строки, начинающиеся с символа решетки (#), игнорируются. Подробности см. на странице руководства audit.conf(5).

11.4. Настройка auditd для среды, защищенной от несанкционированного доступа

Значения по умолчанию auditd должны подходить для большинства окружений. Тем не менее, если окружение в конкретной системе должно отвечать требованиям строгих политик обеспечения безопасности, то мы рекомендуем следующие параметры для демона Audit, настраиваемые в файле /etc/audit/auditd.conf:

log_file

Каталог, содержащий файлы журнала Audit (обычно это /var/log/audit/), должен располагаться на отдельной точке монтирования. Это предотвращает поглощение места в этом каталоге другими процессами и обеспечивает точное определение свободного места, которым располагает демон Audit.

max_log_file

Указывает максимальный размер одного файла журнала Audit, значение должно быть настроено на полное использование доступного места на разделе, на котором располагаются файлы журнала Audit.

max_log_file_action

Определяет, какое действие должно выполняться по достижении лимита, установленного в параметре max_log_file. Здесь необходимо указать keep_logs для предотвращения перезаписи журнала Audit.

space_left

Определяет объем доступного места на диске, для которого будет запускаться действие, указанное в параметре space_left_action. Здесь нужно указать число, которое даст администратору достаточно времени чтобы успеть отреагировать и освободить место на диске. Значение space_left зависит от скорости, с которой создаются записи в журнале Audit.

space_left_action

Для space_left_action рекомендуется указать значение email или exec с соответствующим методом оповещения.

admin_space_left

Указывает абсолютный минимум свободного места на диске, при достижении которого будет запущено действие, указанное для параметра admin_space_left_action. Здесь должно указываться значение, при котором останется достаточно места для журналирования действий, выполняемых администратором.

admin_space_left_action

Здесь нужно указать single, чтобы перевести систему в однопользовательский режим и предоставить администратору возможность освободить свободное место на диске.

disk_full_action

Указывает действие, которое должно запускаться при отсутствии свободного места на разделе, на котором размещены файлы журналов Audit. Здесь нужно указать halt или single. Это обеспечивает выключение системы или работу в однопользовательском режиме, если Audit больше не может выполнять журналирование событий.

disk_error_action

Указывает действие, запускающееся в том случае, если на разделе, на котором размещаются файлы журналов Audit, найдены ошибки. Здесь нужно указать syslog, single или halt в зависимости от локальных параметров политик безопасности относительно обработки сбоев оборудования.

flush

Здесь нужно указать incremental_async. Это значение работает в сочетании с параметром freq, который определяет, какое число записей можно отправить на жесткий диск перед принудительной синхронизацией с ним. Значение для параметра freq должно составлять 100.

Эти параметры обеспечивают синхронизацию данных событий Audit с файлами журнала на диске, одновременно сохраняя хорошую производительность для пиков активности. Остальные параметры должны настраиваться в соответствии с локальными политиками безопасности.

11.5. Запуск службы Audit

Настроив auditd, запускайте службы для начала сбора информации и сохранения ее в файлах журнала. Для запуска auditd выполните следующую команду с привилегиями суперпользователя root:

```
# service auditd start
```

Примечание. Команда service является единственным средством корректного взаимодействия с демоном auditd. Команду service необходимо использовать для правильной записи значения auid. Команду systemctl можно использовать только для двух действий: enable и status.

Чтобы auditd стартовал при загрузке системы, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl enable auditd
```

11.5.1. Определение правил Audit

Система аудита работает на основе набора правил, которые определяют, что именно должно попадать в файлы журнала. Можно выделить следующие типа правил Audit:

Правила управления

Разрешают изменять поведения системы аудит и некоторые из ее параметров

Правила файловой системы

Также известные как «файловые дозоры» («file watches»), они делают возможным наблюдение за доступом к конкретному файлу или каталогу.

Правила системных вызовов

Дают возможность журналирования системных вызовов, которые выполняет любая указанная программа.

Правила аудита можно настроить:

- в командной строке с помощью утилиты auditctl. Обратите внимание, что после перезагрузки эти правила не сохраняются. Подробности см. в параграфе «определение правил Audit с использованием auditctl»;
- в файле /etc/audit/rules.d/audit.rules. Подробности см. в параграфе «определение постоянных правил и средств управления Audit в файле /etc/audit/rules.d/audit.rules».

С помощью команды auditctl можно контролировать базовые действия системы Audit и определять правила, согласно которым Audit выбирает, какие события нужно заносить в журнал.

Примечание. Все команды, взаимодействующие со службой Audit и с файлами журнала Audit, требуют привилегий root. Убедитесь, что запускаете эти команды с привилегиями суперпользователя root. Кроме того, для настройки служб Audit необходим активный параметр ядра CAP_AUDIT_CONTROL, а для журналирования сообщений пользователя — CAP_AUDIT_WRITE.

11.6. Определение правил контроля (управления)

Ниже приводятся некоторые правила контроля, позволяющие изменять поведение системы Audit:

`-b`

Устанавливает максимальное число существующих буферов Audit в ядре, например:

```
# auditctl -b 8192
```

-f

Указывает, какое событие должно выполняться при обнаружении критической ошибки, например:

```
# auditctl -f 2
```

Указанная выше конфигурация в случае критической ошибки запускает kernel panic.

-e

Включает и выключает систему Audit или блокирует ее параметры, например:

```
# auditctl -e 2
```

Указанная выше команда блокирует параметры Audit.

-r

Устанавливает количество создаваемых сообщений в секунду, например:

```
# auditctl -r 0
```

Указанная выше конфигурация убирает количественные ограничения для создаваемых сообщений.

-s

Сообщает статус системы Audit, например:

```
# auditctl -s
AUDIT_STATUS: enabled=1 flag=2 pid=0 rate_limit=0
backlog_limit=8192 lost=259 backlog=0
```

-l

Выводит список всех текущих загруженных правил Audit, например:

```
# auditctl -l
-w /etc/passwd -p wa -k passwd_changes
-w /etc/selinux -p wa -k selinux_changes
-w /sbin/insmod -p x -k module_insertion
⋮
```

-D

Удаляет все загруженные на данный момент правила Audit, например:

```
# auditctl -D
No rules
```

11.6.1. Определение правил файловой системы

Чтобы определить правило файловой системы, используйте следующий синтаксис:

```
auditctl -w <путь_к_файлу> -p <права_доступа> -k <имя_ключа>
```

Здесь:

- `<путь_к_файлу>` — наблюдаемый файл или каталог;
- `<права_доступа>` — журналируемые права:

  r — доступ на чтение для файла или каталога;

  w — доступ на запись для файла или каталога;

x — доступ на выполнение для файла или каталога;

a — изменение в атрибуте файла или каталога.

- <имя_ключа> — это дополнительная строка, помогающая определить, на основе какого правила или набора правил была создана конкретная запись в журнале.

Пример: правило для файловой системы

Чтобы определить правило, на основе которого журналируются все доступы на запись и каждое изменение атрибута файла /etc/passwd, выполните следующую команду:

```
# auditctl -w /etc/passwd -p wa -k passwd_changes
```

Обратите внимание, что строка, следующая за ключом -k, является произвольной.

Чтобы определить правило, на основе которого журналируются все доступы на запись и все изменения атрибутов для всех файлов в каталоге /etc/selinux/, выполните;

```
# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

Чтобы определить правило, на основе которого журналируется выполнение команды /sbin/insmod, вводящей модули в ядро Linux, выполните:

```
# auditctl -w /sbin/insmod -p x -k module_insertion
```

11.6.2. Определение правил для системных вызовов

Чтобы определить правило для системного вызова, используйте следующий синтаксис:

```
auditctl -a <действие>, <фильтр> -S <системный_вызов> -F
<поле>=<значение> -k <имя_ключа>
```

Здесь:

- <действие> и <фильтр> указывают, когда будет журналироваться некоторое событие. <действие> может иметь значение always или never. <фильтр> указывает, какой фильтр ядра, соответствующий правилу, применяется к действию. Фильтром, соответствующим правилу, может быть task, exit, user или exclude. Подробности об этих фильтрах см. в подразделе 11.2. «Архитектура системы Audit»;
- <системный_вызов> указывает на имя системного вызова. Список всех системных вызовов можно найти в файле /usr/include/asm/unistd_64.h. Несколько системных вызовов можно сгруппировать в одном правиле, каждое из них указывается после его собственного параметра -S;
- <поле>=<значение> указывает дополнительные параметры, далее изменяющие правило на соответствие событиям с совпадающей архитектурой, идентификатором группы, идентификатором процесса и другими параметрами. Полный список доступных типов полей и их значений см. на странице руководства auditctl(8);
- <имя_ключа> — это необязательная строка, помогающая определить, на основе какого правила или набора правил была создана конкретная запись в журнале.

Пример: правило системного вызова

Чтобы определить правило, создающее запись в журнале каждый раз, когда программа использует системные вызовы adjtimex или settimeofday в 64-битной системе, выполните следующую команду:

```
# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday
-k time_change
```

Чтобы определить правило, создающее запись в журнале каждый раз, когда системный пользователь со значением ID, равным или более 1000, удаляет или переименовывает файл, выполните:

```
# auditctl -a always,exit -S unlink -S unlinkat -S rename -S
renameat -F auid>=1000 -F auid!=4294967295 -k delete
```

Параметр `-F auid!=4294967295` используется для исключения пользователей без UID.

Также можно определить правило файловой системы с использованием синтаксиса правила системного вызова. Следующая команда создает правило для системных вызовов, аналогичное правилу файловой системы `-w /etc/shadow -p wa`:

```
# auditctl -a always,exit -F path=/etc/shadow -F perm=wa
```

11.6.3. Определение правил для исполняемых файлов

Чтобы определить правило для исполняемых файлов, используйте следующий синтаксис:

```
auditctl -a <действие>, <фильтр> [ -F <архитектура>=<ЦП> -S <си-
стемный_вызов>] -F exe=<путь_до_исполняемого_файла> -k <имя_клю-
ча>
```

Здесь:

- <действие> и <фильтр> указывают, когда будет журналироваться некоторое событие. <действие> может иметь значение always или never. <фильтр> указывает, какой фильтр ядра, соответствующий правилу, применяется к действию. Фильтром, соответствующим правилу, может быть task, exit, user или exclude. Подробности об этих фильтрах см. в подразделе 11.2. «Архитектура системы Audit»;
- <системный_вызов> указывает на имя системного вызова. Список всех системных вызовов можно найти в файле /usr/include/asm/unistd_64.h. Несколько системных вызовов можно сгруппировать в одном правиле, каждое из них указывается после его собственного параметра -S;
- <путь_до_исполняемого_файла> — это абсолютный путь до наблюдаемого выполняемого файла;
- <имя_ключа> — это необязательная строка, помогающая определить, на основе какого правила или набора правил была создана конкретная запись в журнале.

Пример: правило для выполняемого файла

Чтобы определить правило, на основе которого будут журналироваться все выполнения программы /bin/id, выполните следующую команду:

```
# auditctl -a always,exit -S execve -F exe=/usr/bin/tar -k
execution_tar
```

11.7. Настройка постоянных правил и правил управления Audit в файле audit.rules

Чтобы определить правила Audit, сохраняющиеся после перезагрузки системы, их нужно напрямую включить в файл /etc/audit/rules.d/audit.rules. Для указания правил в файле audit.rules используется тот же самый синтаксис консольной команды auditctl. Пустые строки и строки, начинающиеся с символа решетки (#), игнорируются.

С помощью команды auditctl и параметра `-R` также можно читать правила из указанного файла, например:

```
# auditctl -R /usr/share/doc/audit/rules/30-stig.rules
```

11.7.1. Определение правил управления

В файле может содержаться только одно правило, изменяющее поведение системы Audit, из следующего перечня: `-b`, `-D`, `-e`, `-f`, `-r`, `--loginuid-immutable` и `--backlog_wait_time`.

Примеры:

- `-b 8192` — указать размер буфера;
- `-D` — удалить все ранее настроенные правила;
- `-e 2` — сделать параметры неизменяемыми до следующей перезагрузки;
- `-f 2` — при сбое запускать kernel panic;
- `-r 100` — создавать максимум 100 сообщений аудита в секунду;
- `--loginuid-immutable 1` — сделать login UID неизменяемым сразу после его установки (может повредить хранилищам).

11.7.2. Определение правил для файловой системы и системных вызовов

Правила для файловой системы и системных вызовов устанавливаются с использованием синтаксиса auditctl. Примеры из подразделов 11.6.1. «Определение правил файловой системы» и 11.6.2. «Определение правил для системных вызовов» можно представить с помощью следующего файла правил:

```
-w /etc/passwd -p wa -k passwd_changes
-w /etc/selinux/ -p wa -k selinux_changes
-w /sbin/insmod -p x -k module_insertion

-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k
time_change
-a always,exit -S unlink -S unlinkat -S rename -S renameat -F
auid>=1000 -F auid!=4294967295 -k delete
```

11.7.2.1. Файлы предварительно настроенных правил

Пакет audit устанавливает набор предварительно настроенных правил, соответствующих различным стандартам сертификации, в каталоге /usr/share/doc/audit/rules/.

Чтобы использовать эти файлы, создайте резервную копию исходного файла

/etc/audit/audit.rules и сохраните копию нужного файла предварительно настроенных правил под именем /etc/audit/audit.rules:

```
# cp /etc/audit/rules.d/audit.rules
/etc/audit/rules.d/audit.rules_backup
# cp /usr/share/doc/audit/rules/30-stig.rules
/etc/audit/rules.d/audit.rules
```

Примечание. Правила Audit подчиняются схеме нумерации, подробности о которой можно почерпнуть из файла /usr/share/doc/audit/rules/README-rules.

11.8. Использование скрипта augenrules для определения постоянных правил

Скрипт augenrules читает правила, расположенные в каталоге /etc/audit/rules.d/ и объединяет их в файле audit.rules. Этот скрипт обрабатывает все файлы, оканчивающиеся на .rules, в особом порядке на основе их естественного порядка сортировки. Файлы в этом каталоге организованы в группы со следующими значениями:

- 10 — конфигурация для ядра и auditctl;
- 20 — правила, которые могли соответствовать общим правилам, но нам нужны другие соответствия;
- 30 — главные правила;
- 40 — необязательные правила;
- 50 — правила для сервера;
- 70 — локальные системные правила;
- 90 — Finalize (неизменяемые).

Эти правила не предназначены для одновременного использования. Это части политики, которую нужно продумать и затем скопировать отдельные фрагменты в файл /etc/audit/rules.d/. Чтобы, например, настроить конфигурацию STIG, скопируйте правила 10-base-config, 30-stig, 31-privileged и 99-finalize.

Собрав нужные правила в каталоге /etc/audit/rules.d/, загрузите их, выполнив следующий скрипт:

```
# augenrules --load
augenrules --load No rules
enabled 1
failure 1
pid 634
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
enabled 1
failure 1
pid 634
rate_limit 0
```

```
backlog_limit 8192
lost 0
backlog 1
```

Подробности о правилах Audit и скрипте augenrules см. на страницах руководств audit.rules(8) и augenrules(8).

11.9. Чтение файлов журнала Audit

По умолчанию система Audit хранит записи журнала в файле /var/log/audit/audit.log; при включенной ротации файлов журнала файлы audit.log хранятся в том же каталоге.

Следующее правило заносит в журнал каждую попытку чтения или изменения файла /etc/ssh/sshd_config:

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

При работающем демоне auditd следующая команда создаст, например, новое событие в файле журнала Audit:

```
$ cat /etc/ssh/sshd_config
```

В файле audit.log это событие выглядит следующим образом:

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e
syscall=2 success=no exit=-13 a0=7fffd19c5592 a1=0
a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=1000
uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000
sgid=1000 fsgid=1000 tty=pts0 ses=1 comm="cat" exe="/bin/cat"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="sshd_config"
type=CWD msg=audit(1364481363.243:24287): cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0
name="/etc/ssh/sshd_config" inode=409248 dev=fd:00 mode=0100600
ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
type=PROCTITLE msg=audit(1364481363.243:24287) :
proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

Вышеуказанное событие состоит из четырех записей с одинаковой меткой времени и серийным номером. Записи всегда начинаются с `type=keyword`. Каждая запись состоит из нескольких пар `<имя>=<значение>`, разделенных пробелами или запятыми.

11.9.1. Поиск по файлам журнала Audit

Утилита ausearch предоставляет возможность поиска конкретных событий в файлах журнала Audit. По умолчанию ausearch выполняет поиск в файле /var/log/audit/audit.log. С помощью команды ausearch <параметры> -if <имя_файла> можно указать другой файл. Указание нескольких параметров в одной команде ausearch равнозначно использованию оператора AND между типами полей и оператора OR между несколькими вхождениями одного и того же типа поля.

Пример: использование ausearch для поиска по файлам журнала Audit

Чтобы найти в файле /var/log/audit/audit.log неудачные попытки входа в систему, выполните следующую команду:

```
# ausearch --message USER_LOGIN --success no --interpret
```

Чтобы найти все изменения в учетной записи, изменения группы и роли, выполните:

```
# ausearch -m ADD_USER -m DEL_USER -m ADD_GROUP -m USER_CHAUTHTOK
-m DEL_GROUP -m CHGRP_ID -m ROLE_ASSIGN -m ROLE_REMOVE -i
```

Чтобы найти все входы в систему, осуществленные конкретным пользователем с пользовательским ID (auid), выполните:

```
# ausearch -ua 1000 -i
```

Чтобы найти все неудачные системные вызовы со вчерашнего дня по настоящее время:

```
# ausearch --start yesterday --end now -m SYSCALL -sv no -i
```

Полный список всех возможностей ausearch см. на странице руководства ausearch(8).

Пример: создание отчета с помощью утилиты aureport

Как можно было убедиться в подразделе 11.9. «Чтение файлов журнала Audit», файлы журнала Audit записываются в машиночитаемом формате. Для создания на их основе отчетов и удобства чтения администратором существует утилита aureport.

Утилита aureport предназначена для создания сводки и отчетов, напечатанных столбцами, о событиях, записанных в файлах журнала Audit. По умолчанию для создания отчета утилита обращается к файлам в каталоге /var/log/audit/. Чтобы указать другой файл, на основе которого нужно создать отчет, выполните команду с ключом `-if <имя_файла>`.

Пример: использование aureport для создания отчета Audit

Чтобы создать отчет на основе событий, заносимых в журнал аудита в течение последних трех дней, исключая текущий, выполните следующую команду:

```
# aureport --start 04/08/2017 00:00:00 --end 04/11/2017 00:00:00
```

Чтобы создать отчет о всех событиях исполнения файлов, выполните:

```
# aureport -x
```

Чтобы создать сводку на базе отчета о событиях исполняемых файлов, полученного выше, выполните:

```
# aureport -x --summary
```

Чтобы создать сводный отчет о сбойных событиях для всех пользователей, выполните:

```
# aureport -u --failed --summary -i
```

Чтобы создать сводный отчет всех неудачных попытках входа для каждого системного пользователя, выполните:

```
# aureport --login --summary -i
```

Чтобы создать отчет на базе поиска ausearch всех событий доступа к файлам для пользователя с ID 1000, выполните:

```
# ausearch --start today --loginuid 1000 --raw | aureport -f
--summary
```

Чтобы создать отчет обо всех запрошенных файлах Audit и о временном промежут-

ке, в который происходили журналируемые в них события, выполните:

```
# aureport -t
```

Полный список всех возможностей aureport см. на странице руководства aureport(8).

Документация и страницы руководств по системе Audit, которые можно найти в установленной ОС РОСА «КОБАЛЬТ»:

Документацию пакета audit можно найти в каталоге /usr/share/doc/audit-2.6.5/.

Страницы руководств:

- audispd.conf(5);
- auditd.conf(5);
- ausearch-expression(5);
- audit.rules(7);
- audispd(8);
- auditctl(8);
- auditd(8);
- aulast(8);
- aulastlog(8);
- aureport(8);
- ausearch(8);
- ausyscall(8);
- autrace(8);
- auvirt(8).

11.10. Аудит системы: практические примеры

1. Должна осуществляться идентификация и проверка подлинности субъектов доступа при входе в систему по паролю условно-постоянного действия длиной не менее 8 (восьми) буквенно-цифровых символов.

Выполнение проверки:

1) Задать правила паролей (минимум 8 буквенно-цифровых символов), для этого в файле /etc/pam.d/password-auth добавить в конец строки

   ```
   password    requisite     pam_cracklib.so try_first_pass retry=3 type=
   ```

   следующие параметры:

   ```
   dcredit=-1 lcredit=-1 minlen=8
   ```

2) Создать пользователя user с произвольным паролем. Для этого понадобится повысить права до учетной записи суперпользователя root.
3) Войти в ОС пользователем user, попытаться сменить пароль в текстовой консоли с помощью команды passwd на:

   а) пароль длиной менее 8 символов,

   б) пароль, состоящий только из цифр,

в) пароль, состоящий только из букв,

г) корректный пароль из не менее чем 8 буквенно-цифровых символов.

4) Убедиться, что пользователь не может использовать пароль, не удовлетворяющий требованиям политики безопасности.
5) Войти в ОС созданным пользователем, введя корректный пароль данного пользователя.
6) Повысить права до учетной записи суперпользователя root и провести анализ файла журнала безопасности /var/log/secure на предмет наличия в нем данных идентификации данного пользователя.

Анализ файла журнала можно провести, используя утилиты командной строки, такие как cat или tail, или графический редактор текстовый файлов Leafpad.

Вы должны увидеть запись вида:

```
Jun 21 15:18:23 cobaltsx2 lxdm-binary: pam_unix(lxdm:session):
session opened for user user by (uid=0)
```

2. Должна осуществляться идентификация терминалов, ЭВМ, узлов сети ЭВМ по логическим именам.

Данные по идентификации узла, проводящего подключение к данной ОС, можно посмотреть в логах подключения.

Для идентификации узлов сети по логическим именам необходимо настроить службу DNS и использовать статическую IP-адресацию. В этом случае обращение к любому сетевому узлу будет осуществляться по его логическому имени.

При этом, если имела место попытка доступа по тому или иному протоколу, всегда есть возможность провести идентификацию обратившегося объекта командой nslookup.

Например, при подключении по протоколу ssh в логе /var/log/secure при помощи следующей команды:

```
tail -f /var/log/secure
```

...вывод лога будет таким:

```
Jun 28 08:59:19 srv1 sshd[22396]: Accepted publickey for user
from 192.168.10.1 port 59894 ssh2
```

Чтобы идентифицировать объект с IP-адресом 192.168.10.1, нужно выполнить команду:

```
nslookup 192.168.10.1
```

...которая выдаст логическое имя узла сети в виде:

```
1.10.168.192.in-addr.arpa       name = srv1.test.ru.
```

3. Должна осуществляться регистрация входа (выхода) субъектов доступа в систему (из системы).

В параметрах регистрации указываются:

- дата и время входа (выхода) субъекта доступа в систему (из системы);
- результат попытки входа: успешная или неуспешная (несанкционированная);
- идентификатор (код или фамилия) субъекта, предъявленный при попытке доступа.

Выполнение проверки:

1) Войти в ОС ранее созданным пользователем user, введя корректный пароль данного пользователя.
2) Выйти из системы.
3) Попытаться войти в ОС под созданным пользователем, введя некорректный пароль для данного пользователя.
4) Попытаться войти в ОС под не существующим в системе пользователем, введя произвольный пароль.
5) Провести анализ файла журнала безопасности /var/log/secure на предмет наличия в нем параметров регистрации входа/выхода с нужными параметрами.

Пример журнала регистрации:

```
Jun 22 15:04:51 cobaltsx1 lxdm-binary: pam_unix(lxdm:session):
session opened for user ozherelev by (uid=0)
```

4. Должна осуществляться регистрация запуска (завершения) программ и процессов (заданий, задач), предназначенных для обработки защищаемых файлов.

В параметрах регистрации указываются:

- дата и время запуска;
- имя (идентификатор) программы (процесса, задания);
- идентификатор субъекта доступа, запросившего программу (процесс, задание);
- результат запуска (успешный, неуспешный (несанкционированный).

Данный пункт требований выполняется благодаря системе аудита auditd. Это механизм, имеющий широкие возможности настройки.

Чтобы включить контроль запуска приложений, нужно в конец файла /etc/audit/audit.rules добавить сточку следующую строку:

```
-a exit,always -F arch=b64 -S execve -k EXE
```

Далее перезапустите сервис аудита:

```
service restart auditd
```

Теперь в файл аудита /var/log/audit/audit.log будут добавляться события запуска любого приложения. Можно провести операцию запуска сервиса Postgresql или любого другого.

Посмотреть события можно с помощью команды ausearch, также имеющей массу фильтров. Если вы хотите посмотреть события, связанные с защищаемым файлом, выполните следующую команду:

```
ausearch -k EXE
```

Если вас, например, интересуют события за определенное время:

```
ausearch -k EXE --start 06/15/16 10:00 --end 06/15/16 12:00
```

Например, запуск СУБД Postresql будет выглядеть так:

```
time->Tue Jun 21 16:05:12 2016
type=PATH msg=audit(1466514312.938:9311): item=1 name=(null)
inode=263134 dev=fc:02 mode=0100755 ouid=0 ogid=0 rdev=00:00
nametype=NORMAL
type=PATH msg=audit(1466514312.938:9311): item=0
```

```
name="/usr/pgsql-9.5/bin/postmaster" inode=280558 dev=fc:02
mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL
type=CWD msg=audit(1466514312.938:9311): cwd="/var/lib/pgsql"
type=EXECVE msg=audit(1466514312.938:9311): argc=3
a0="/usr/pgsql-9.5/bin/postmaster" a1="-D"
a2="/var/lib/pgsql/9.5/data"
type=SYSCALL msg=audit(1466514312.938:9311): arch=c000003e
syscall=59 success=yes exit=0 a0=1007930 a1=1007890 a2=1016e50
a3=7fff575d8260 items=2 ppid=1 pid=3912 auid=500 uid=26 gid=26
euid=26 suid=26 fsuid=26 egid=26 sgid=26 fsgid=26 tty=(none)
ses=1059 comm="postmaster" exe="/usr/pgsql-9.5/bin/postgres"
key="EXE"
```

5. ФБО должны быть способны генерировать запись аудита для следующих событий, потенциально подвергаемых аудиту:

5.1. Обработка отказов аутентификации

- достижение ограничения для неуспешных попыток аутентификации и соответствующие предпринятые действия (например, отключение терминала), а также, при необходимости, последующие действия, направленные на восстановление нормального состояния (например, предоставление возможности заново работать с терминалом);

Должны регистрироваться дата и время события, тип события, идентификатор субъекта и результат события (успешный или неуспешный).

Отказы аутентификации записываются в файл /var/log/secure, где и могут быть проанализированы, например, с помощью команд tail или cat.

Для настройки ограничения по неуспешным попыткам доступа необходимо в файл /etc/pam.d/system-auth и /etc/pam.d/password-auth добавить две строки:

- в секцию auth:

```
auth        required      pam_tally2.so  file=/var/log/tallylog
deny=3 even_deny_root unlock_time=1200
```

Здесь deny=3 задает количество неуспешных попыток входа пользователя до его блокировки;

`unlock_time` — время, на которое блокируется пользователь;

- в секцию account:

```
account     required      pam_tally2.so
```

Посмотреть данные по неуспешным входам пользователей можно с помощью команды `pam_tally2`.

Проведение проверки

Пользователь user при попытке входа в ОС вводит некорректный пароль три раза, после чего пытается войти в ОС, введя корректный пароль. ОС должна блокировать его вход, при этом в файле журнала /var/log/secure должны появиться записи вида:

```
Jun 22 12:46:52 srv1 lxdm-binary: pam_tally2(lxdm:auth): user
user (500) tally 3, deny 3
```

5.2. Управление атрибутами безопасности

- все изменения значений атрибутов безопасности.

Должны регистрироваться дата и время события, тип события, идентификатор субъекта и результат события (успешный или неуспешный).

Атрибуты безопасности ОС задаются в различных конфигурационных файлах системы в каталоге /etc. Для аудита изменения значений атрибутов нужно установить аудит на данные файлы.

Предположим, что вы хотите вести аудит изменений конкретного защищаемого файла /boot/grub/grub.conf (настойка опций загрузки ОС). Для этого внесите следующую строку в конец файла /etc/audit/audit.rules:

```
-w /boot/grub/grub.conf -p rwa -k MON_GRUB
```

Далее перезапустите сервис аудита следующей командой:

```
/etc/init.d/auditd restart
```

Теперь в файл аудита /var/log/audit/audit.log будут добавляться события чтения, записи и изменения атрибутов данного файла.

Проведение проверки

Выполните операции чтения и изменения данного файла.

Посмотреть события можно с помощью команды ausearch, также имеющей массу фильтров. Если вы хотите посмотреть события, связанные с защищаемым файлом, выполните следующую команду:

```
ausearch -k MON_GRUB
```

Вывод данной команды будет выглядеть следующим образом:

```
time->Wed Jun 22 12:52:00 2016
type=PATH msg=audit(1466589120.363:3381): item=0
name="/boot/grub/grub.conf" inode=658691 dev=fc:02 mode=0100600
ouid=0 ogid=0 rdev=00:00 nametype=NORMAL
type=CWD msg=audit(1466589120.363:3381): cwd="/boot/grub"
type=SYSCALL msg=audit(1466589120.363:3381): arch=c000003e
syscall=89 success=no exit=-22 a0=7fff38c01ff0 a1=7fff38c02ff0
a2=fff a3=7fff38c03ff0 items=1 ppid=4503 pid=4580 auid=500 uid=0
gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=3
comm="mc" exe="/usr/bin/mc" key="MON_GRUB"
```

6. Обработка отказов аутентификации. ФБО должны обнаруживать, когда произойдет определенное уполномоченным администратором число неуспешных попыток аутентификации

Число неуспешных попыток аутентификации задается опцией «deny» в файле /etc/pam.d/system-auth в следующей строке:

```
auth          required       pam_tally2.so   file=/var/log/tallylog
deny=3 even_deny_root unlock_time=1200
```

Здесь `deny=3` задает количество неуспешных попыток входа пользователя до его блокировки,

`unlock_time` — время, на которое блокируется пользователь. Если убрать параметр `unlock_time`, пользователь не сможет авторизоваться, пока администратор вруч-

ную не сбросит количество неуспешных попыток входа.

Узнать текущее количество неуспешных попыток входа можно с помощью команды `pam_tally2`. Подробное описание модуля pam_tally2 приведено на man-странице pam_tally2.

12. МЕЖСЕТЕВОЙ ЭКРАН FIREWALLD

Демон firewalld предоставляет динамически управляемый сетевой экран с поддержкой сетевых «зон» для присвоения уровня доверия сети и связанным с ней соединениям и интерфейсам. Firewalld поддерживает параметры IPv4 и IPv6, мосты Ethernet и IP set, а постоянные параметры конфигураций в нем отделены от текущей рабочей среды. Также имеется интерфейс для прямого добавления правил сетевого экрана для служб или приложений. Вся связь с сетевым экраном выполняется с помощью D-Bus.

12.1. Введение

Для ускорения действий, изменяющих набор правил, демон firewalld по умолчанию использует команды `restore` из iptables, ip6tables и ebtables. Обычные команды используются, если в файле firewalld.conf для параметра IndividualCalls указано значение `yes` или же если используется запасное решение, когда правила нельзя применить с помощью команд `restore`. Применение обычных команд значительно замедляет работу.

Служба межсетевого экрана, предоставляемая firewalld, является скорее динамической, чем статической, поскольку изменения в конфигурации можно внести в любой момент, и они тут же начинают действовать. Необходимость сохранять или применять изменения отсутствует. Непреднамеренный разрыв существующих сетевых соединений невозможен, т. к. ни одна из составных частей сетевого экрана не нуждается в перезагрузке.

В составе программы поставляется консольный клиент firewall-cmd. С его помощью можно вносить постоянные и непостоянные динамические изменения во время работы, что объясняется на man-странице firewall-cmd(1). Постоянные изменения нужно применять так, как это объясняется на странице руководства firewalld(1). Обратите внимание, что команда firewall-cmd может выполняться как пользователем root, так и пользователем-администратором, иными словами, членом группы wheel. В последнем случае команда проходит авторизацию через механизм polkit.

Для внесения изменений в постоянное окружение консольный клиент firewall-offline-cmd может использоваться только пользователем root. Клиент не общается напрямую с firewalld, но использует часть firewalld core и внутренние процессы ввода/вывода для изменения конфигурации. Не рекомендуется работать с этим инструментом при активном сетевом экране. Его можно использовать, но изменения, внесенные с помощью firewall-offline-cmd, не сразу применяются к firewalld. Изменения в постоянном окружении будут применены после того, как firewalld обнаружит изменения в файлах ФС. Команду firewall-offline-cmd используют во время установки ОС для настройки межсетевого экрана; также ее можно использовать на этапе сразу после установки для изменения параметров межсетевого экрана перед загрузкой только что установленной системы.

Приложение firewall-applet может быстро запустить окно параметров NetworkManager для используемых сетевых соединений. Внести изменения в присвоенную зону межсетевого экрана можно на вкладке «Общее». Это приложение по умолчанию не устанавливается в ОС РОСА «КОБАЛЬТ». Его можно установить следующей командой:

```
yum install firewall-applet
```

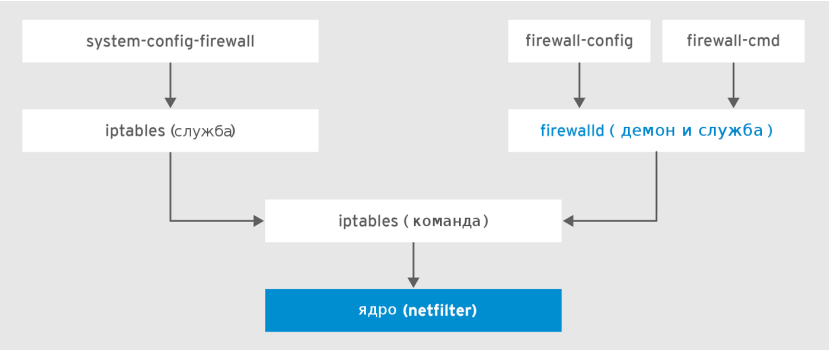
Параметры firewalld хранятся в нескольких файлах формата XML в каталогах

/usr/lib/firewalld/ и /etc/firewalld, что предоставляет значительную гибкость в управлении, т. к. файлы можно редактировать, в них можно писать, можно сохранять их резервные копии, использовать их в качестве шаблонов для других установок и т. д. Конфигурация в каталоге /usr/lib/firewalld/ является конфигурацией по умолчанию, а конфигурация в каталоге /etc/firewalld/ является частной конфигурацией конкретной системы.

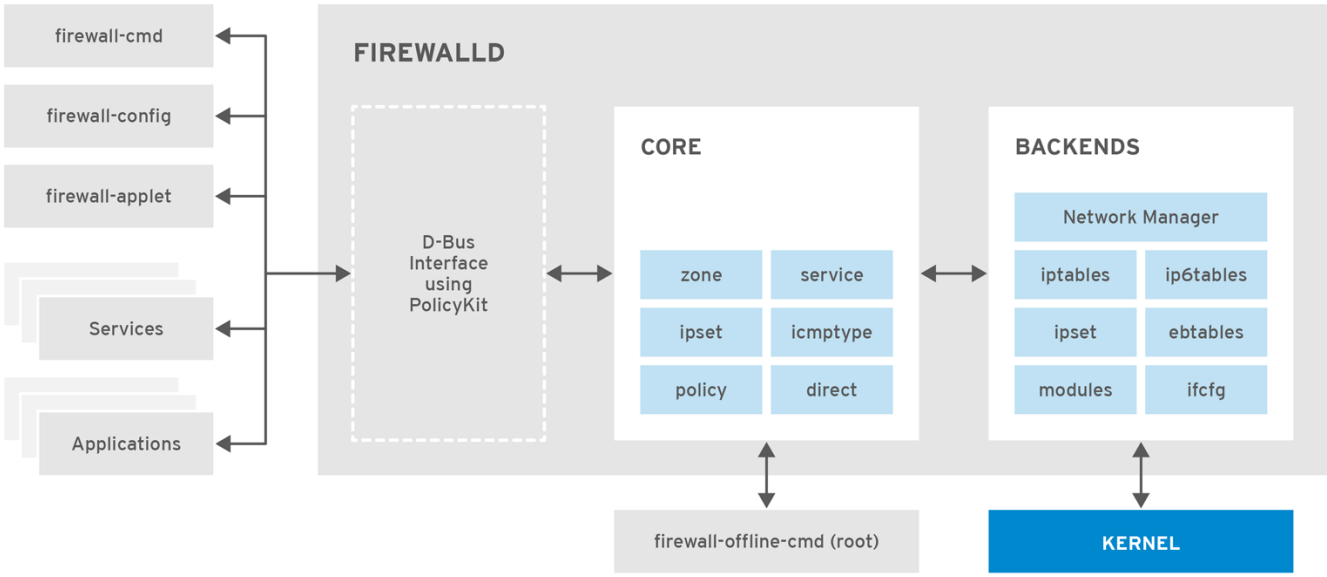
Все приложения обмениваются информацией с firewalld при помощи интерфейса D-Bus.

Стек и архитектура работы сервиса firewalld показаны на рисунках.

Примечание. Firewalld не может импортировать параметры межсетевого экрана из файлов /etc/sysconfig/ip*tables. Для импорта параметров lokkit или system-config-firewall используйте firewall-offline-cmd и файл /etc/sysconfig/system-config-firewall. Файлы пользовательских правил нельзя импортировать в firewalld. Импортируемые параметры применяются к зоне по умолчанию.



Стек межсетевого экрана



Архитектура firewalld

12.2. Сетевые зоны

Firewalld используется для разделения сетей на различные зоны на базе уровня доверия пользователя к интерфейсам и трафику внутри сети. NetworkManager информирует firewalld о том, к какой зоне принадлежит интерфейс. Зону, присвоенную интерфейсу, мож-

но изменить с помощью NetworkManager или с помощью утилиты firewall-config. Если интерфейс контролируется программой NetworkManager, а пользователь изменит зону этого интерфейса с помощью firewall-cmd, firewall-offline-cmd или firewall-config, этот запрос будет перенаправлен программе NetworkManager, и демон firewalld не будет его обрабатывать.

Параметры зон в каталоге /etc/firewalld/ представляют собой диапазон предварительно установленных значений, которые можно быстро применить к сетевому интерфейсу. Эти параметры коротко описываются ниже.

- drop — все входящие сетевые пакеты сбрасываются без ответа. Возможны только исходящие соединения;
- block — все исходящие сетевые соединения отклоняются с сообщением icmp-host-prohibited для IPv4 и icmp6-adm-prohibited для IPv6. Разрешаются только сетевые соединения, инициированные внутри системы;
- public — для работы в зонах общего использования. Компьютеры в сети рассматриваются как потенциальные источники угроз; принимаются только конкретные входящие соединения;
- external — для использования во внешних сетях с включенным маскарадингом, особенно для роутеров. Компьютеры в сети рассматриваются как потенциальные источники угроз; принимаются только конкретные входящие соединения;
- dmz — для машин в собственной «демилитаризованной зоне» с общим доступом и ограниченным доступом во внутреннюю сеть. Принимаются только конкретные входящие соединения;
- work — для использования в рабочих зонах. Машины в сети рассматриваются как не представляющие угрозы. Принимаются только конкретные входящие соединения;
- home — для использования дома. Машины в сети рассматриваются как не представляющие угрозы. Принимаются только конкретные входящие соединения;
- internal — для использования во внутренних сетях. Машины в сети рассматриваются как не представляющие угрозы. Принимаются только конкретные входящие соединения;
- trusted — принимаются все сетевые соединения.

Одну из этих зон можно назначить зоной по умолчанию. При добавлении интерфейсных подключений в NetworkManager они присваиваются зоне по умолчанию. При установке системы зоной по умолчанию в firewalld назначается зона public.

12.3. Выбор сетевой зоны

Зонам присвоены понятные названия, позволяющие пользователям быстро сделать выбор. В соответствии с конкретными потребностями и оценками рисков администратор должен изучить параметры по умолчанию и отключить ненужные службы.

Имена зон и значения параметров являются рекомендациями, которые можно изменить в соответствии с потребностями. Встроенную зону нельзя удалить, но можно отка-

тить параметры зоны к исходным значениям по умолчанию, загрузив их в составе постоянной конфигурации firewall-config или firewall-cmd.

12.4. Предварительно настроенные службы

Служба может представлять собой:

1) Список локальных портов, протоколов, исходных портов и точек назначения.
2) Список вспомогательных модулей межсетевого экрана, автоматически загружаемых при активации службы.

Использование предварительно настроенных служб облегчает пользователю включение и отключение доступа к службам. Использование предварительно настроенных служб или служб, настроенных пользователем, в противовес открытию портов или диапазонов портов, облегчает выполнение административных задач. Параметры настройки служб и общая информация о файлах приведены на странице руководства firewalld.service(5). Службы конкретизируются с помощью отдельных конфигурационных файлов ввида <имя_службы>.xml. В firewalld имена протоколов предпочтительнее имен служб или приложений.

Чтобы просмотреть список доступных в системе служб, выполните следующую команду:

```
$ firewall-cmd --get-services
```

Чтобы просмотреть параметры службы, выполните:

```
$ firewall-cmd --info-service=<имя_службы>
```

Чтобы получить список только предварительно настроенных доступных служб, выполните:

```
$ ls /usr/lib/firewalld/services/
```

Примечание. Для просмотра списка файлов в /usr/lib/firewalld не требуются привилегии суперпользователя root. После добавления частных пользовательских файлов не забудьте соответствующим образом изменить их атрибуты.

Файлы в каталоге /usr/lib/firewalld/services/ редактировать запрещено; можно редактировать только файлы из каталога /etc/firewalld/services/. Чтобы получить список системных служб или служб, созданных пользователем, выполните следующую команду с привилегиями суперпользователя root:

```
# ls /etc/firewalld/services/
```

Добавлять и удалять службы можно с помощью графической утилиты firewall-config, а также утилит firewall-cmd и firewall-offline-cmd. Кроме того, можно редактировать файлы XML в каталоге /etc/firewalld/services/. Если пользователь не добавлял иди не изменял службу, соответствующий XML-файл будет отсутствовать в каталоге /etc/firewalld/services/. Если требуется добавить или изменить службу, файлы из /usr/lib/firewalld/services/ можно использовать в качестве шаблонов.

Чтобы добавить новую службу в консоли, используйте `firewall-cmd` или, если межсетевой экран неактивен, `firewall-offline-cmd`. Для добавления новой пустой службы выполните следующую команду:

```
$ firewall-cmd --permanent --new-service=<имя_службы>
```

Для добавления новой службы на базе локального файла выполните:

```
$ firewall-cmd --permanent --new-service-from-file=<имя_служ-
бы>.xml
```

Имя службы можно изменить с помощью дополнительного параметра `--name=<имя_службы>`

Сразу после изменения параметров службы обновленная копия службы помещается в каталог /etc/firewalld/services/. Чтобы скопировать службу вручную, выполните:

```
# cp /usr/lib/firewalld/services/service-name.xml
/etc/firewalld/services/service-name.xml
```

Firewalld в первую очередь загружает службы из каталога /usr/lib/firewalld/services. Если файлы размещаются в /etc/firewalld/services и отвечают требованиям, указанные в них значения будут иметь приоритет над аналогичными файлами из /usr/lib/firewalld/services. Файлы из /usr/lib/firewalld/services будут использованы сразу после того, как совпадающие файлы будут удалены из /etc/firewalld/services или как демону firewalld будет указано загрузить значения этой службы по умолчанию. Эти правила применяются только к постоянному окружению. Для отката параметров в работающем окружении также требуется перезагрузка.

12.5. Прямой интерфейс

Прямой интерфейс firewalld позволяет напрямую передавать правила сервисам iptables, ip6tables и ebtables. В первую очередь он предназначен для использования приложениями. Неопытным администраторам, слабо знающим iptables, не рекомендуется использовать прямой интерфейс, т. к. это может привести к появлению уязвимостей в межсетевом экране. Во время использования составных частей отслеживаемого интерфейса можно попросить firewalld показать изменения, сделанные приложением, использующим этот режим. Неотслеживаемый сквозной (passthrough) режим предназначен только для служб, полностью контролирующих свой набор правил, таких как libvirt и docker. Прямой интерфейс реализуется добавлением параметра `--direct` команде `firewall-cmd`.

Режим прямого интерфейса предназначен для добавления конкретных правил межсетевого экрана службами или приложениями в работающем режиме. Правила можно сделать постоянными, выполнив команду `firewall-cmd --direct` с ключом `--permanent` или же внеся изменения в файл /etc/firewalld/direct.xml. Если правила не были сделаны постоянными, их необходимо применять каждый раз, когда от firewalld через D-Bus приходит сообщение «start», «restart» или «reload». Прямые правила также можно использовать в цепочках конкретных зон.

12.6. Работа с firewalld

12.6.1. Установка firewalld

В ОС РОСА «КОБАЛЬТ» firewalld устанавливается по умолчанию. При необходимости в этом можно убедиться, выполнив следующую команду с привилегиями суперпользователя root:

```
# yum install firewalld
```

Чтобы установить графический инструмент настройки firewall-config, выполните:

```
# yum install firewall-config
```

Чтобы установить дополнительный firewall-applet, выполните:

```
# yum install firewall-applet
```

12.6.2. Остановка firewalld

Чтобы остановить работу firewalld, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl stop firewalld
```

Чтобы предотвратить автоматический запуск firewalld при загрузке системы, выполните:

```
# systemctl disable firewalld
```

Чтобы убедиться, что firewalld не будет стартовать при доступе к интерфейсу D-Bus, а также в случае, если firewalld необходим другим службам, выполните:

```
# systemctl mask firewalld
```

12.6.3. Запуск firewalld

Для запуска firewalld выполните следующие команды с привилегиями суперпользователя root:

```
# systemctl unmask firewalld
# systemctl start firewalld
```

Для автоматического запуска firewalld при загрузке системы выполните:

```
# systemctl enable firewalld
```

12.6.4. Проверка статуса firewalld

Чтобы проверить, запущен ли firewalld, выполните следующую команду:

```
$ systemctl status firewalld
```

При запущенном сервисе вывод данной команды будет выглядеть так:

```
 firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service;
enabled; vendor preset: enabled)
   Active: active (running) since Tue 2016-10-11 09:15:58 CEST; 2
days ago
     Docs: man:firewalld(1)
 Main PID: 721 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─721 /usr/bin/python -Es /usr/sbin/firewalld --nofork
--nopid

Oct 11 09:15:57 localhost.localdomain systemd[1]: Starting
firewalld - dynami...
Oct 11 09:15:58 localhost.localdomain systemd[1]: Started
firewalld - dynamic...
```

```
Hint: Some lines were ellipsized, use -l to show in full.
```

Дополнительно, чтобы проверить связь firewall-cmd с демоном, выполните:

```
$ firewall-cmd —state
```

При запущенном сервисе вывод данной команды будет выглядеть так:

```
   running
```

12.7. Настройка firewalld

Службу firewall, реализованную в виде демона firewalld, можно настроить с помощью графического инструмента firewall-config и консольных утилит firewall-cmd и firewall-offline-cmd, а также путем редактирования конфигурационных файлов в формате XML.

12.7.1. Настройка firewall с помощью консольной утилиты firewall-cmd

Консольная утилита firewall-cmd является частью приложения firewalld, устанавливаемой по умолчанию. Проверить ее наличие в системе можно, проверив версию или запросив вывод справки при помощи параметра `--help`.

Для проверки версии выполните:

```
$ firewall-cmd --version
```

Для вывода справки выполните:

```
$ firewall-cmd --help
We list a selection of commands below; for a full list see the
firewall-cmd(1) man page.
```

Примечание. Чтобы сделать результаты команды долговременными или постоянными, добавляйте ко всем командам (кроме команд с ключом `--direct`, которые по своей природе являются временными) параметр `--permanent`. Обратите внимание, что это означает не только то, что изменения будут постоянными, но и то, что изменения вступят в силу только после перезагрузки firewalld, перезапуска службы или перезагрузки всей системы. Изменения, внесенные с помощью команды `firewall-cmd` без параметра `--permanent`, вступают в силу немедленно, но действуют только до следующей перезагрузки firewall, системной перезагрузки или перезапуска службы firewalld. Перезагрузка firewalld сама по себе не прерывает соединений. Но не забывайте, что при этом сбрасываются все временные изменения.

Чтобы результаты команды применились немедленно и также стали постоянными, выполните команду дважды: один раз с ключом `--permanent` и один раз — без него. Это необходимо, поскольку перезагрузка firewalld занимает больше времени, чем повторное выполнение команды, из-за необходимости перезагрузки всех файлов конфигурации межсетевого экрана. Во время перезагрузки политика для встроенных цепочек меняется на DROP из соображений безопасности, а в конце меняется на ACCEPT. Во время процесса перезагрузки возможны перерывы в обслуживании.

Примечание. Все изменения привязки зон к интерфейсам, находящимся под контролем NetworkManager, перенаправляются программе NetworkManager. Если запрос к NetworkManager успешен, эти изменения не применяются к конфигурации firewalld. Аналогичная ситуация происходит и при использовании параметра `--permanent`.

Для интерфейсов, не находящихся под контролем NetworkManager, изменение применяется к конфигурации firewalld. Если интерфейс используется файлом ifcfg, параметр `ZONE=` этого файла адаптируется для обеспечения однородности конфигурации firewalld и файла ifcfg. Если интерфейс используется несколькими файлами ifcfg, тогда используется первый из них.

Для таких параметров конфигурации, как зона по умолчанию, при использовании консольных и графических утилит разница между работающим окружением и постоянным окружением отсутствует.

12.8. Просмотр параметров межсетевого экрана в консольном режиме

Чтобы получить текстовую информацию о состоянии firewalld, выполните следующую команду:

```
$ firewall-cmd --state
```

Чтобы просмотреть список активных зон со списком присвоенных им в данный момент интерфейсов, выполните:

```
$ firewall-cmd --get-active-zones
public
  interfaces: em1
```

Чтобы узнать, какой зоне присвоен интерфейс, например, em1, выполните:

```
$ firewall-cmd --get-zone-of-interface=em1
Public
```

Чтобы узнать все интерфейсы, присвоенные зоне, например, зоне public, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=public --list-interfaces
          em1 wlan0
```

Эта информация получается от NetworkManager и показывает только интерфейсы, но не соединения.

Чтобы узнать параметры, например, зоны public, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=public --list-all
public
  interfaces:
  services: mdns dhcpv6-client ssh
  ports:
  forward-ports:
  icmp-blocks: source-quench
```

Чтобы узнать информацию по зоне, используйте ключ `--info-zone`. Чтобы получить подробный вывод с описаниями и короткими описаниями, дополнительно используйте ключ `-v`.

```
# firewall-cmd --info-zone=public
public (active)
  target: default
```

```
  icmp-block-inversion: no
  interfaces: em1
  sources:
  services: dhcpv6-client mdns ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Чтобы просмотреть список текущих загруженных служб, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --get-services
cluster-suite pop3s bacula-client smtp ipp radius bacula ftp mdns
samba dhcpv6-client dns openvpn imaps samba-client http https ntp
vnc-server telnet libvirt ssh ipsec ipp-client amanda-client
tftp-client nfs tftp libvirt-tls
```

Здесь перечисляются названия предварительно настроенных служб, загруженных из /usr/lib/firewalld/services/, а также всех пользовательских служб, запруженных в настоящее время. Обратите внимание, что сами конфигурационные файлы имеют названия вида <имя_службы>.xml.

Чтобы получить список служб, созданных, но еще не загруженных пользователем, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --permanent --get-services
```

Здесь перечислены все службы, включая службы, настроенные пользователем в /etc/firewalld/services/, даже если они еще не загружены.

Чтобы просмотреть параметры службы ftp, выполните:

```
# firewall-cmd --info-service=ftp
ftp
  ports: 21/tcp
  protocols:
  source-ports:
  modules: nf_conntrack_ftp
  destination:
```

Чтобы просмотреть параметры в режиме постоянной конфигурации, используйте ключ `--permanent`.

12.9. Изменение параметров межсетевого экрана в консольном режиме

12.9.1. Сброс всех пакетов (режим паники)

Чтобы начать сброс всех входящих и исходящих пакетов, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --panic-on
```

Все входящие и исходящие пакеты будут сброшены. Активные соединения будут закрыты после периода неактивности; время на выполнение этой операции зависит от установленных значений времени ожидания в конкретном сеансе.

Чтобы снова начать передачу входящих и исходящих пакетов, выполните:

```
# firewall-cmd --panic-off
```

После отключения режима паники установленные соединения снова могут заработать, если режим паники продлился недолгое время. Чтобы узнать, включен или выключен режим паники, выполните:

```
$ firewall-cmd --query-panic
```

При включенном режиме эта команды выведет «yes» со статусом выхода 0. В противном случае будет выведено «no» со статусом выхода 1.

12.9.2. Перезагрузка firewalld

Чтобы перезагрузить межсетевой экран, не прерывая соединений пользователей (без потери информации о состоянии), выполните следующую команду:

```
$ firewall-cmd --reload
```

Процесс перезагрузки межсетевого экрана включает в себя перезагрузку всех конфигурационных файлов и повторное создание всей конфигурации firewall. Во время перезагрузки политика для встроенных цепочек из соображений безопасности меняется на DROP, а в конце меняется на ACCEPT. Соответственно, во время перезагрузки возможны перерывы в обслуживании. Как вариант, с привилегиями суперпользователя root пошлите сигнал SIGHUP для перезагрузки межсетевого экрана.

Для перезагрузки межсетевого экрана и прерывания соединений пользователей со сбросом информации о состоянии, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --complete-reload
```

Обычно эту команду применяют только в случае серьезных проблем в межсетевом экране. Используйте ее, например, при наличии проблем с информацией о состоянии, когда соединение не удается установить при корректных правилах межсетевого экрана.

12.9.3. Добавление интерфейса в зону

Чтобы добавить интерфейс в зону (например, интерфейс em1 в зону public), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=public --add-interface=em1
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.4. Добавление интерфейса в зону путем редактирования конфигурационного файла

Чтобы добавить интерфейс в зону (например, интерфейс em1 в зону work), получите привилегии суперпользователя root и добавьте в файл ifcfg-em1 следующую строку:

```
ZONE=work
```

Обратите внимание, что если опустить значение параметра `ZONE` или использовать `ZONE=` или `ZONE=''`, будет использована зона по молчанию.

NetworkManager автоматически выполнит повторное соединение, и зона будет соответствующим образом настроена.

12.9.5. Установка параметров зоны по умолчанию путем редактирования конфигурационного файла

С привилегиями суперпользователя root откройте файл /etc/firewalld/firewalld.conf и внесите в него следующие изменения:

```
# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=home
```

Перезагрузите firewall, выполнив следующую команду:

```
# firewall-cmd --reload
```

Это действие перезагрузит межсетевой экран без потери информации о состоянии (то есть сеансы TCP не будут прерваны), но во время перезагрузки возможен отказ в обслуживании.

12.9.6. Установка зоны по умолчанию

Чтобы установить зону по умолчанию (например, public), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --set-default-zone=public
```

Это изменение вступит в силу мгновенно; перезагружать межсетевой экран не нужно.

12.9.7. Открытие портов

Чтобы получить список всех открытых портов в зоне (например, в dmz), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=dmz --list-ports
```

Обратите внимание, что эта команда не покажет порты, открытые в результате выполнения команды с ключом `--add-services`.

Чтобы добавить порт в зону (например, разрешить трафик TCP для порта 8080 в зоне dmz), выполните:

```
# firewall-cmd --zone=dmz --add-port=8080/tcp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

Чтобы добавить диапазон портов в зону (например, разрешить порты с 5060 до 5061 в зоне public), выполните:

```
# firewall-cmd --zone=public --add-port=5060-5061/udp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.8. Открытие протоколов в консольном режиме

Чтобы получить список всех открытых портов в зоне (например, в зоне dmz), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=dmz --list-protocols
```

Обратите внимание, что эта команда не покажет протоколов, открытых в результате выполнения команды `firewall-cmd --add-services`.

Чтобы добавить протокол в зону (например, разрешить трафик ESP в зоне dmz), выполните:

```
# firewall-cmd --zone=dmz --add-protocol=esp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.9. Открытие портов-источников в консольном режиме

Чтобы получить список всех открытых портов-источников в зоне (например, в зоне dmz), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=dmz --list-source-ports
```

Обратите внимание, что эта команда не покажет портов-источников, открытых в результате выполнения команды `firewall-cmd —add-services`.

Чтобы добавить порт-источник в зону (например, разрешить трафик TCP из порта 8080 в зоне dmz), выполните:

```
# firewall-cmd --zone=dmz --add-source-port=8080/tcp
```

Чтобы добавить диапазон портов-источников в зону (например, разрешить порты с 5060 до 5061 для зоны public), выполните:

```
# firewall-cmd --zone=public --add-source-port=5060-5061/udp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.10. Добавление службы в зону в консольном режиме

Чтобы добавить службу в зону (например, разрешить SMTP для зоны work), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=work --add-service=smtp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.11. Удаление службы из зоны в консольном режиме

Чтобы удалить службу из зоны (например, удалить SMTP из зоны work), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=work --remove-service=smtp
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

Это изменение не разорвет уставленных соединений, но если это необходимо, используйте ключ `--complete-reload`. Учтите, что это действие разорвет все соединения не только для удаляемой службы.

12.9.12. Добавление службы в зону путем редактирования файлов XML

Чтобы просмотреть список файлов для зоны по умолчанию, выполните следующую команду с привилегиями суперпользователя root:

```
# ls /usr/lib/firewalld/zones/
block.xml  drop.xml      home.xml      public.xml   work.xml
dmz.xml    external.xml  internal.xml  trusted.xml
```

Эти файлы нельзя редактировать. Они используются по умолчанию в случае отсутствия эквивалентных файлов в каталоге /etc/firewalld/zones/.

Для просмотра файлов зон, отличающихся от файлов по умолчанию, выполните следующую команду с привилегиями суперпользователя root:

```
# ls /etc/firewalld/zones/
external.xml  public.xml  public.xml.old
```

В примере выше файла зоны work не существует. Чтобы добавить файл зоны work, выполните:

```
# cp /usr/lib/firewalld/zones/work.xml /etc/firewalld/zones/
```

Теперь можно редактировать файл в каталоге /etc/firewalld/zones/. В случае его удаления firewalld вернется к использованию файла по умолчанию в каталоге /usr/lib/firewalld/zones/.

Чтобы добавить службу в зону (например, разрешить SMTP для зоны work), добавьте следующую строку в файл /etc/firewalld/zones/work.xml с привилегиями суперпользователя root:

```
<service name="smtp"/>
```

12.9.13. Удаление службы из зоны помощью редактирования файлов XML

Для редактирования файлов зоны в формате XML необходим текстовый редактор, запущенный с привилегиями суперпользователя root.

Чтобы просмотреть файлы для ранее настроенных зон, выполните следующую команду:

```
# ls /etc/firewalld/zones/
external.xml  public.xml  work.xml
```

Чтобы удалить службу из зоны (например, службу SMTP из зоны work), отредактируйте файл /etc/firewalld/zones/work.xml, удалив следующую строку:

```
<service name="smtp"/>
```

Если в файл work.xml не было внесено других изменений, его можно удалить, и firewalld будет использовать файл по умолчанию /usr/lib/firewalld/zones/work.xml после следующей своей перезагрузки или перезагрузки всей системы.

12.9.14. Настройка маскарадинга IP-адресов

Чтобы проверить, включен ли маскарадинг IP-адресов (например, для внешней зоны), выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=external --query-masquerade
```

В случае включенного маскарадинга команда выдаст «yes» ос статусом выхода 0. В противном случае ответ будет «no» со статусом выхода 1. Если зона не указана, используется зона по умолчанию.

Чтобы включить маскарадинг, выполните:

```
# firewall-cmd --zone=external --add-masquerade
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

Чтобы отключить маскарадинг, выполните:

```
# firewall-cmd --zone=external --remove-masquerade
```

Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

12.9.15. Настройка проброса портов

Чтобы перенаправить входящие сетевые пакеты с одного порта на другой либо на адрес, сначала нужно включить проброс портов в зоне (например, в зоне external). Выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --zone=external --add-masquerade
```

Чтобы перенаправить пакеты на локальный порт (расположенный в той же самой системе), выполните:

```
# firewall-cmd --zone=external —add-forward-port=
port=22:proto=tcp:toport=3753
```

В этом примере пакеты, предназначенные для порта 22, теперь направляются на порт 3753. Исходный порт назначения задается ключом `port`, с помощью которого можно указать как порт, так и диапазон портов. Аналогично указывается протокол. В качестве протокола нужно указать `tcp` или `udp`. Новый локальной порт (порт или диапазон портов, на который перенаправляется трафик), указывается ключом `toport`. Чтобы сделать это изменение постоянным, повторно выполните команду с ключом `--permanent`.

Чтобы перенаправить пакеты на другой адрес IPv4 (обычно это внутренний адрес) без изменения порта назначения, выполните:

```
# firewall-cmd --zone=external —add-forward-port=
port=22:proto=tcp:toaddr=192.0.2.55
```

В этом примере пакеты, предназначенные для порта 22, теперь перенаправляются на тот же порт по адресу, указанному ключом `toaddr`. Все остальные ключи работают аналогично тем, что использованы в примере выше.

Чтобы перенаправить пакеты на другой порт и другой адрес IPv4 (обычно это внутренний адрес), выполните:

```
# firewall-cmd --zone=external \
```

```
      --add-forward-
port=port=22:proto=tcp:toport=2055:toaddr=192.0.2.55
```

В этом примере пакеты, предназначенные для порта 22, теперь перенаправляются на порт 2055 по адресу, указанному ключом `toaddr`. Все остальные ключи работают аналогично тем, что использованы в примере выше.

12.10. Настройка firewall с помощью файлов XML

Параметры настройки firewalld хранятся в файлах XML в каталоге /etc/firewalld/. Не изменяйте файлы в каталоге /usr/lib/firewalld/ (эти файлы определяют значения по умолчанию). Для просмотра и редактирования файлов XML необходимы привилегии суперпользователя root. Структура этих файлов описывается на трех страницах руководств:

- man-страница firewalld.icmptype(5) — конфигурационные файлы в формате XML, относящиеся к фильтрации трафика ICMP;
- man-страница firewalld.service(5) — описываются конфигурационные файлы в формате XML, относящиеся к службе firewalld;
- man-страница firewalld.zone(5) — описываются конфигурационные файлы в формате XML, относящиеся к настройке зон firewalld.

12.10.1. Использование прямого интерфейса

Ключ `--direct` утилиты firewall-cmd делает возможным добавление и удаление цепочек в рабочем режиме. Ниже приведены несколько примеров того, как это можно сделать. Подробности см. на странице руководства firewall-cmd(1).

Неопытным администраторам, слабо знающим iptables, не рекомендуется использовать прямой интерфейс, т. к. это может привести к появлению уязвимостей в межсетевом экране. Режим прямого интерфейса предназначен для добавления службам или приложениям конкретных правил firewall в рабочем режиме. Правила можно сделать постоянными, выполнив команду `firewall-cmd --direct` с ключом `--permanent` или же внеся изменения в файл /etc/firewalld/direct.xml. Подробную информацию о файле /etc/firewalld/direct.xml см. на странице руководства firewalld.direct(5).

12.10.1.1. Добавление правила с использованием прямого интерфейса

Чтобы добавить правило в цепочку «IN_public_allow», выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --direct --add-rule ipv4 filter IN_public_allow \
        0 -m tcp -p tcp --dport 666 -j ACCEPT
```

Чтобы сделать это значение постоянным, укажите ключ `--permanent`.

12.10.1.2. Удаление правила с использованием прямого интерфейса

Чтобы удалить правило из цепочки «IN_public_allow», выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --direct --remove-rule ipv4 filter IN_public_allow
\
        0 -m tcp -p tcp --dport 666 -j ACCEPT
```

Чтобы сделать это значение постоянным, укажите ключ `--permanent`.

12.10.1.3. Просмотр правил с помощью прямого интерфейса

Чтобы просмотреть правила в цепочке «IN_public_allow», выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --direct --get-rules ipv4 filter IN_public_allow
```

Обратите внимание, что эта команда (ключ `--get-rules`) показывает только те правила, которые ранее были добавлены с помощью ключа `--add-rule`. Она не показывает правила iptables, добавленные с помощью других средств.

12.10.2. Создание сложных правил firewall с использованием синтаксиса «rich language»

С помощью синтаксиса «rich language» можно создавать сложные правила межсетевого экрана более простым и понятным способом, нежели использование прямого интерфейса. Кроме того, эти параметры можно сделать постоянными. Язык использует ключевые слова и является абстрактным представлением правил iptables. С помощью этого языка можно настраивать зоны.

Примеры см. на странице руководства firewalld.richlanguage(5).

12.10.3. Блокировка межсетевого экрана

При запуске с правами root локальные приложения или службы (например, libvirt) могут изменить параметры межсетевого экрана. Но администратор может заблокировать параметры firewall так, чтобы запретить всем приложениям запрашивать изменения в межсетевом экране, а также разрешить эти запросы только тем приложениям, которые добавлены в белый список блокировки. По умолчанию параметры блокировки отключены. При их включении администратор может быть уверен, что в параметры межсетевого экрана не будут внесены изменения со стороны локальных приложений или служб.

12.10.4. Настройка блокировки firewall

Запустите текстовый редактор с привилегиями суперпользователя root и добавьте следующую строчку в файл /etc/firewalld/firewalld.conf:

```
Lockdown=yes
```

Перезагрузите межсетевой экран:

```
# firewall-cmd --reload
```

Попробуйте включить службу imaps в зоне по умолчанию с помощью следующей команды от имени пользователя-администратора (пользователя, состоящего в группе wheel. Обычно это первый пользователь, созданный в системе). Будет выведен запрос на ввод пароля этого пользователя:

```
$ firewall-cmd --add-service=imaps
Error: ACCESS_DENIED: lockdown is enabled
```

Чтобы разрешить использование firewall-cmd, выполните:

```
# firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python
-Es /usr/bin/firewall-cmd*'
```

Чтобы сделать это значение постоянным, укажите ключ `--permanent`.

Перезагрузите межсетевой экран с привилегиями суперпользователя root:

```
# firewall-cmd --reload
```

Попробуйте снова включить службу imaps в зоне по умолчанию:

```
$ firewall-cmd –add-service=imaps
```

Вам потребуется ввести пароль пользователя. В этот раз команда должна быть успешно выполнена.

12.11. Использование службы iptables

Чтобы использовать службы iptables и ip6tables вместо firewalld, сначала отключите firewalld. Для этого выполните следующие команды с привилегиями суперпользователя root:

```
# systemctl disable firewalld
# systemctl stop firewalld
```

Далее установите пакет iptables-services:

```
# yum install iptables-services
```

Пакет iptables-services содержит службу iptables и службу ip6tables.

Чтобы запустить службы iptables и ip6tables, выполните:

```
# systemctl start iptables
# systemctl start ip6tables
```

Чтобы включить старт служб при каждом запуске системы, выполните:

```
# systemctl enable iptables
# systemctl enable ip6tables
```

Установленная локально документация по firewalld:

- firewalld(1) — командные параметры firewalld;
- firewalld.conf(5) — информация о конфигурации firewalld;
- firewalld-applet(1) — параметры утилиты firewall-applet;
- firewall-cmd(1) — командные параметры консольного клиента firewalld;
- firewall-config(1) — параметры инструмента firewall-config;
- firewall-offline-cmd(1) — командные параметры клиента offline;
- firewalld.icmptype(5) — описание конфигурационных файлов в формате XML для фильтрации ICMP;
- firewalld.ipset(5) — описание конфигурационных файлов в формате XML для firewalld IP sets;
- firewalld.service(5) — описание конфигурационных файлов в формате XML для службы firewalld;
- firewalld.zone(5) — описание конфигурационных файлов в формате XML для зон firewalld;
- firewalld.direct(5) — описание конфигурационного файла прямого интерфейса firewalld;

- firewalld.lockdown-whitelist(5) — описание конфигурационного файла белого списка блокировки firewall;
- firewall.richlanguage(5) — описание синтаксиса правил rich language;
- firewalld.zones(5) — общее описание зон и установки их параметров;
- firewalld.dbus(5) — описание интерфейса D-Bus для firewalld.

13. БАЗОВАЯ НАСТРОЙКА ПОСЛЕ УСТАНОВКИ В СЕРВЕРНОМ ВАРИАНТЕ

В этом разделе будут рассмотрены установка и настройка базовых сервисов, необходимых для работы сервера. Предполагается, что ОС была установлена в конфигурации пакетов «Минимальная».

Большая часть настроек будет выполняться в командной строке от имени суперпользователя root. Там, где настройка должна происходить от иного пользователя, это будет указано отдельно.

В ОС семейства Linux большая часть параметров самой операционной системы и сервисов, в ней работающих, хранятся в обычных текстовых конфигурационных файлах. Таким образом, настройка ОС и ее сервисов по большей части сводится к редактированию соответствующих файлов с последующим их перечитыванием (reload) нужным сервисом. Не все параметры могут быть просто перечитаны; иногда сервис приходится полностью перезагружать (restart).

13.1. Просмотр статуса сетевых соединений

Посмотреть статус текущих сетевых соединений можно следующей командой:

```
ip a
```

Если вы привыкли к использованию утилиты *ifconfig*, нужно установить пакет, содержащий данную утилиту, т. к. в минимальной установке он отсутствует. Для этого нужно смонтировать установочный DVD в каталог /mnt. Сделать это можно следующей командой:

```
mount /dev/sr0 /mnt/
```

Если вы проводили установку с иного носителя, например, USB, примонтируйте соответствующий носитель в каталог /mnt.

По умолчанию в только что установленной ОС РОСА «КОБАЛЬТ» подключен только один репозиторий для установки пакетов — DVD, с которого вы проводили установку.

Чтобы установить пакет, содержащий утилиту *ifconfig*, выполните следующую команду:

```
yum -y install net-tools
```

Посмотреть текущие сетевые соединения можно с помощью команды `ifconfig`.

13.2. Настройка сетевых соединений

Данная настройка может производиться как с помощью команды *ifconfig* (временно — при перезапуске ОС данная настройка будет потеряна), так и путем редактирования файлов конфигурации.

В первом случае временная настройка сетевого интерфейса выполняется следующими командами:

- `dhclient eth0` (для автоматической настройки интерфейса по протоколу DHCP);
- `ifconfig eth0 192.168.1.1 netmask 255.255.255.0` (в случае статической конфигурации IP).

Рассмотрим подробнее второй способ.

Файлы настройки сетевого интерфейса обычно называются /etc/sysconfig/network-scripts/ifcfg-<имя>, где <имя> относится к имени устройства, управляемого этим конфигурационным файлом.

Редактировать файлы можно при помощи любого текстового редактора, например, vi, vim или mcedit.

13.2.1. Настройка получения IP-адреса по DHCP

Настройка сети по протоколу DHCP предусматривает использование значения BOOTPROTO=dhcp. Конфигурационный файл при этом будет выглядеть так:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

После изменения конфигурационного файла необходимо перезапустить сетевую службу:

```
systemctl restart network
```

Настройка сети со статическим IP-адресом предусматривает использование значения BOOTPROTO=none, а также указания в явном виде сетевых параметров. Конфигурационный файл при этом будет выглядеть так:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
DNS1=192.168.1.100
```

После изменения конфигурационного файла необходимо перезапустить сетевую службу:

```
systemctl restart network
```

13.2.2. Настройка соединения типа VLAN

Создайте файл конфигурации нового соединения VLAN, например, /etc/sysconfig/network-scripts/ifcfg-vlan1000, со следующим содержанием:

```
ONBOOT=yes
TYPE=Ethernet
VLAN=yes
VLAN_NAME_TYPE=VLAN_PLUS_VID_NO_PAD
DEVICE=vlan1000
PHYSDEV=eth0
VLAN_ID=1000
BOOTPROTO=static
IPADDR=192.168.1.1
NETMASK=255.255.255.0
```

В данном примере была создана VLAN с ID=1000.

`vlan_name_type`

После изменения конфигурационного файла необходимо перезапустить сетевую службу:

```
systemctl restart network
```

Параметр `vlan_name_type` задает тип именования интерфейса VLAN. Он может принимать следующие значения:

`VLAN_PLUS_VID` — `vlan01000`
`VLAN_PLUS_VID_NO_PAD` — `vlan1000`
`DEV_PLUS_VID` — `eth0.01000`
`DEV_PLUS_VID_NO_PAD` — `eth0.1000`

14. УПРАВЛЕНИЕ СЛУЖБАМИ С ПОМОЩЬЮ SYSTEMD

14.1. Введение

Systemd — это средство управления системой и службами для операционных систем Linux. Systemd была разработана с учетом обратной совместимости со сценариями, инициализации SysV, и предлагает такие возможности, как параллельный запуск системных служб во время загрузки системы, активация демонов по требованию, поддержка снимков ОС, а также логику управления службами на основе зависимостей. В ОС РОСА «КОБАЛЬТ» systemd заменила Upstart в качестве системы инициализации по умолчанию.

Systemd вводит понятие системных юнитов. Эти юниты представлены конфигурационными файлами, расположенными в одном или нескольких каталогах, и в сжатом виде аккумулируют информацию о системных службах, слушающих сокетах, сохраненных снимках состояния системы, а также о других объектах, имеющих отношение к системе инициализации. Полный список доступных типов юнитов systemd см. в таблице 1.

Таблица 1. Доступные типы юнитов systemd

Тип юнита	Расширение файла	Описание
Служба	.service	Системная служба
Цель	.target	Группа юнитов systemd
Автоматическое монтирование	.automount	Точка автоматического монтирования файловой системы
Устройство	.device	Файл устройства, распознаваемого ядром
Монтирование	.mount	Точка монтирования файловой системы
Путь	.path	Файл или каталог в файловой системе
Область	.scope	Процесс, созданный извне
Срез	.slice	Группа иерархически организованных юнитов, управляющих системными процессами
Снимок	.snapshot	Сохраненное состояние менеджера systemd
Сокет	.socket	Сокет для обмена информацией между процессами
Подкачка	.swap	Устройство или файл подкачки
Таймер	.timer	Таймер systemd

Таблица 2. Местоположение файлов юнитов systemd

Каталог	Описание
/usr/lib/systemd/system/	Файлы юнитов systemd, распределенные установленными пакетами RPM

/run/systemd/system/	Файлы юнитов systemd, созданные в процессе работы. Этот каталог имеет приоритет над каталогом с установленными служебными файлами юнитов
/etc/systemd/system/	Файлы юнитов systemd, созданные командой `systemctl enable`, а также файлы юнитов, добавленные для расширения службы. Этот каталог имеет приоритет над каталогом с файлами юнитов, созданными в процессе работы.

Переопределение параметров systemd по умолчанию с помощью файла system.conf

Конфигурация systemd по умолчанию определяется во время компиляции, и ее можно просмотреть в файле /etc/systemd/system.conf. Если нужно изменить параметры по умолчанию и глобально переопределить отдельные значения юнитов systemd, используйте этот файл.

Чтобы, например, переопределить значение для предела времени ожидания, по умолчанию равное 90 секундам, измените параметр `DefaultTimeoutStartSec`:

```
DefaultTimeoutStartSec=<требуемое_значение_в_секундах>
```

Также см. пример «Изменение предела истечения времени ожидания».

14.2. Основные возможности

В ОС РОСА «КОБАЛЬТ» системы systemd и service manager предлагают следующие основные возможности:

1) **Активация на основе сокетов**. Во время загрузки systemd создает слушающие сокеты для всех системных служб, поддерживающих этот тип активации, и передает эти сокеты соответствующим службам, как только эти службы начнут работу. Это не только дает systemd возможность параллельного запуска служб, но также делает возможным перезапуск служб без потери сообщений, посланных им в то время, пока они были недоступны: соответствующий сокет остается доступным, и все сообщения помещаются в очередь.

   Для активации на базе сокетов systemd использует *юниты сокетов*.

2) **Активация на основе шины**. Системные службы, использующие D-Bus для обмена информацией между процессами, можно запустить по требованию, как только клиентское приложение в первый раз попробует обменяться с ними информацией. Для активации на основе шины systemd использует файлы службы D-Bus.

3) **Активация на основе устройств**. Системные службы, поддерживающие активацию на основе устройств, можно запустить по требованию при подключении определенного типа оборудования или если это оборудования становится доступно. Для активации на основе устройств systemd использует юниты устройств.

4) **Активация на основе путей**. Системные службы, поддерживающие активацию на основе путей, могут быть запущены по требованию, когда определенный файл или каталог сменят свой статус. Для активации на основе путей systemd использует

юниты пути.

5) **Снимки состояния системы**. systemd может временно сохранить текущее состояние всех юнитов или повторно активировать предыдущее состояние системы из динамически созданного снимка.
6) **Управление точками монтирования и автомонтирования**. systemd управляет точками монтирования и автоматического монтирования. Для точек монтирования systemd использует юниты точек монтирования, а для точек автомонтирования — юниты автомонтирования.
7) **Агрессивная параллелизация**. Благодаря использованию активации на основе сокетов systemd может запускать системные службы параллельно, как только слушающие сокеты окажутся на месте. В сочетании с системными службами, поддерживающими активацию по требованию, параллельная активация значительно сокращает время, требуемое для загрузки системы.
8) **Транзакционная логика активации юнитов**. Перед активацией или отключением юнита systemd рассчитывает его зависимости, создает временную транзакцию и проверяет целостность этой транзакции. Если транзакция будет нецелостной, systemd перед тем, как выдать сообщение об ошибке, автоматически попытается скорректировать транзакцию и удалить из нее задачи, не имеющие критического значения.
9) **Обратная совместимость с системой инициализации SysV**. systemd поддерживает сценарии инициализации SysV, что описано в базовой спецификации проекта Linux Standard Base. Это облегчает переход к служебным юнитам systemd.

14.3. Изменения совместимости

Системы systemd и service manager разработаны так, чтобы обеспечивать базовую совместимость с SysV и Upstart. Далее описываются наиболее значительные изменения в совместимости относительно ОС РОСА «КОБАЛЬТ» и ряда предыдущих дистрибутивов:

1) systemd имеет ограниченную поддержку уровней выполнения, предоставляя несколько юнитов цели, которые возможно напрямую сопоставить с этими уровнями выполнения. Также из соображений совместимости в состав systemd была включена команда `runlevel`. Тем не менее, не все цели systemd возможно напрямую сопоставить с уровнями выполнения, и поэтому команда `runlevel` может вернуть N для обозначения неизвестного уровня выполнения. Рекомендуется по возможности избегать использования команды `runlevel`. Подробную информацию о работе с целями systemd и их сравнение с уровнями выполнения см. в п. 14.4.11. «Работа с целями systemd».
2) Утилита systemctl не поддерживает произвольные команды. В дополнение к стандартным командам, таким, как `start`, `stop` и `status`, авторы сценариев инициализации SysV смогли реализовать поддержку для любого числа произвольных команд в качестве дополнительной функциональности. Например, в Red Hat Enterprise Linux 6 сценарий init для iptables можно было выполнять внутри команды `panic`, что немедленно включало режим panic и перенастраивало систему на немедленный сброс всех входящих и исходящих пакетов. Ничего из этого не поддерживается

в systemd, и systemctl принимает только команды, перечисленные в документации. Подробности об утилите systemctl и сравнение ее с прежней утилитой service см. в п. 14.4. «Управление системными службами».

3) Утилита systemctl не обменивается информацией со службами, которые не были запущены systemd. Когда systemd запускает системную службу, идентификатор ее главного процесса сохраняется для возможности его отслеживания. Затем утилита systemctl использует этот PID для опроса и управления службой. Соответственно, если пользователь запускает определенный демон напрямую из командной строки, systemctl не в состоянии определить его текущий статус или остановить его выполнение.
4) Systemd останавливает только работающие службы. Ранее после инициации последовательности shutdown ОС РОСА «КОБАЛЬТ» и более ранние релизы системы использовали символьные ссылки, расположенные в каталоге /etc/rc0.d/, для остановки всех доступных системных служб вне зависимости от их статуса. С systemd во время выполнения последовательности shutdown останавливаются только работающие службы.
5) Системные службы не могут читать из стандартного потока input. Когда systemd запускает службу, стандартный ввод службы подключается к /dev/null для предотвращения любого взаимодействия с пользователем.
6) Системные службы не наследуют контекста (такого, как переменные окружения HOME и PATH) от вызывающего их пользователя или его сеанса. Каждая служба работает в чистом контексте выполнения.
7) Во время загрузки сценария инициализации SysV systemd читает информацию о зависимостях, закодированную в заголовке Linux Standard Base (LSB) и интерпретирует ее во время процесса работы.
8) Для предотвращения зависания системы по вине неправильно работающей службы все операции с юнитами служб подпадают под действие лимита времени ожидания, равного 5 минутам. Это значение жестко настроено для всех служб, создаваемых сценариями инициации, и его нельзя изменить. Тем не менее, для увеличения лимита времени ожидания на каждую службу можно использовать индивидуальные конфигурационные файлы. См. пример «Изменение лимита времени ожидания» на стр. 183.

14.4. Управление системными службами

В предыдущих версиях ОС РОСА, в которых использовались SysV init или Upstart, сценарии инициализации хранились в каталоге /etc/rc.d/init.d/. Как правило, эти сценарии были написаны на Bash и давали системным администраторам возможность управления состоянием служб или демонов в системе. В ОС РОСА «КОБАЛЬТ» эти сценарии инициализации были заменены на юниты служб.

Юниты служб имеют расширение .service и служат для тех же самых целей, что и сценарии инициализации. Чтобы просмотреть, запустить, остановить, перезапустить, включить или отключить системные службы, используйте команду `systemctl`. Работа с ней описана в таблице «Сравнение утилит service и systemctl», в таблице «Сравнение

утилит chkconfig и systemctl» и далее в этом подразделе. Команды `service` и `chkconfig` все еще доступны в системе и работают, как ожидается, но включены только из соображений совместимости, и их использования следует избегать.

Таблица 3. Сравнение утилит service и systemctl

service	systemctl	Описание
`service <имя> start`	`systemctl start <имя>.service`	Запускает службу
`service <имя> stop`	`systemctl stop <имя>.service`	Останавливает службу
`service <имя> restart`	`systemctl restart <имя>.service`	Перезапускает службу
`service <имя> condrestart`	`systemctl try-restart <имя>.service`	Перезапускает службу, только если она выполняется
`service <имя> reload`	`systemctl reload <имя>.service`	Перезагружает конфигурацию
`service <имя> status`	`systemctl status <имя>.service`	Проверяет, выполняется ли служба
`service --status-all`	`systemctl list-units --type service --all`	Показывает статус всех служб

Таблица 4. Сравнение утилит chkconfig и systemctl

chkconfig	systemctl	Описание
`chkconfig <имя> on`	`systemctl enable <имя>.service`	Включает службу
`chkconfig <имя> off`	`systemctl disable <имя>.service`	Отключает службу
`chkconfig --list <имя>`	`systemctl status <имя>.service` `systemctl is-enabled <имя>.service`	Проверяет, включена ли служба
`chkconfig --list`	`systemctl list-unit-files --type service`	Выводит список всех служб и проверяет, включены ли они
`chkconfig --list`	`systemctl list-dependencies --after`	Выводит список служб, которые должны запускаться раньше указанного юнита
`chkconfig --list`	`systemctl list-dependencies --before`	Выводит список служб, которые должны запускаться после указанного юнита

14.4.1. Указание юнитов служб

Для ясности во всех примерах команд далее в этом пункте используются полные имена юнитов с расширением .service, например:

```
# systemctl stop nfs-server.service
```

Тем не менее, расширение файла можно опускать, и в этом случае утилита systemctl подразумевает, что в качестве аргумента указан юнит службы. Команда ниже является эквивалентом команды в примере выше:

```
# systemctl stop nfs-server
```

Кроме того, у некоторых юнитов есть имена-псевдонимы, которые могут быть короче, чем начальные имена, что удобно для использования. Чтобы узнать все псевдонимы, которые можно использовать для конкретного юнита, выполните:

```
# systemctl show nfs-server.service -p Names
```

14.4.2. Поведение systemctl в окружении chroot

При смене корневого каталога с помощью команды chroot большинство команд systemctl отказываются выполнять какие-либо действия. Причина этому — процесс systemd и пользователь, вызвавший команду chroot, видят разное представление файловой системы. Это происходит, например, когда systemctl вызывается из файла kickstart.

Исключением являются такие команды файла юнита, как `systemctl enable` и `systemctl disable`. Эти команды не влияют на выполняющиеся процессы, но они влияют на файлы юнитов. Поэтому эти команды можно выполнять даже в окружении chroot. Чтобы, например, включить службу httpd в системе в каталоге /srv/website1/, выполните:

```
# chroot /srv/website1
# systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-
user.target.wants/httpd.service, pointing to
/usr/lib/systemd/system/httpd.service.
```

14.4.3. Получение списка служб

Чтобы получить список загруженных на данный момент служб, выполните следующую команду:

```
$ systemctl list-units --type service
```

Для каждого файла юнита службы эта команда покажет его полное имя (`UNIT`), примечание о том, был ли файл юнита загружен (`LOAD`), статус активации файла юнита высокого (`ACTIVE`) и низкого уровня (`SUB`), а также короткое описание (`DESCRIPTION`).

По умолчанию команда `systemctl list-units` отображает только активные юниты. Чтобы просмотреть все загруженные юниты независимо от их статуса, выполните эту команду с ключом `--all` или `-a`:

```
$ systemctl list-units --type service --all
```

Также можно получить список всех доступных юнитов служб, чтобы узнать, включены ли они. Для этого выполните:

```
$ systemctl list-unit-files --type service
```

Для каждого юнита службы эта команда показывает его полное имя (`UNIT FILE`) и данные о том, включен ли юнит службы или нет (`STATE`). Информацию о том, как определить статус отдельных юнитов служб, см. в п. 14.4.4. «Просмотр статуса службы».

Пример: получение списка служб

Чтобы просмотреть список всех загруженных на данный момент юнитов служб, выполните следующую команду:

```
$ systemctl list-units --type service

UNIT                              LOAD   ACTIVE SUB     DESCRIPTION
abrt-ccpp.service                 loaded active exited  Install ABRT
coredump hook
abrt-oops.service                 loaded active running ABRT kernel
log watcher
abrt-vmcore.service               loaded active exited  Harvest
vmcores for ABRT
abrt-xorg.service                 loaded active running ABRT Xorg
log watcher
abrtd.service                     loaded active running ABRT
Automated Bug Reporting Tool
...
systemd-vconsole-setup.service loaded active exited  Setup
Virtual Console
tog-pegasus.service               loaded active running OpenPegasus
CIM Server

LOAD   = Reflects whether the unit definition was properly
loaded.
ACTIVE = The high-level unit activation state, i.e.
generalization of SUB.
SUB    = The low-level unit activation state, values depend on
unit type.

46 loaded units listed. Pass --all to see loaded but inactive
units, too.
To show all installed unit files use 'systemctl list-unit-files'
```

Чтобы получить список всех установленных файлов юнитов служб и узнать, включены ли они, выполните:

```
$ systemctl list-unit-files --type service
UNIT FILE                                   STATE
abrt-ccpp.service                           enabled
abrt-oops.service                           enabled
abrt-vmcore.service                         enabled
abrt-xorg.service                           enabled
abrtd.service                               enabled
```

```
...
wpa_supplicant.service                          disabled
ypbind.service                                  disabled

208 unit files listed.
```

14.4.4. Просмотр статуса службы

Чтобы просмотреть подробную информацию о юните службы, соответствующем системной службе, выполните следующую команду:

```
$ systemctl status <имя>.service
```

Замените `<имя>` именем юнита службы, которую нужно просмотреть (например, gdm). Данная команда отобразит имя выбранной службы, ее короткое описание, одно или несколько полей, указанных ниже в таблице «Доступная информация о юницах служб», и, в случае запуска команды с привилегиями суперпользователя root, также недавние записи в журнале службы.

Таблица 5. Доступная информация о юнитах служб

Поле	Описание
Loaded	Была ли загружена служба, абсолютный путь до файла юнита, а также примечание о том, включен ли юнит.
Active	Выполняется ли юнит службы, а также указывается метка времени
Main PID	PID соответствующей системной службы, а также указывается имя службы
Status	Дополнительная информация о соответствующей системной службе
Process	Дополнительная информация о связанных процессах
CGroup	Дополнительная информация о связанных контрольных группах (cgroups).

Чтобы только проверить, выполняется ли конкретная служба, выполните:

```
$ systemctl is-active <имя>.service
```

Чтобы определить, включен ли юнит конкретной службы, выполните:

```
$ systemctl is-enabled <имя>.service
```

Обратите внимание, что обе команды возвращают статус выхода 0, если указанный юнит службы выполняется или включен. Сведения о том, как получить список всех загруженных на данный момент юнитов, см. в п. 14.4.3. «Получение списка служб».

Пример: просмотр статуса службы

Юнит службы для менеджера графического входа в систему GDM называется gdm.service. Чтобы определить текущий статус этого юнита службы, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl status gdm.service
```

```
gdm.service - GNOME Display Manager
   Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled)
   Active: active (running) since Thu 2013-10-17 17:31:23 CEST;
5min ago
 Main PID: 1029 (gdm)
   CGroup: /system.slice/gdm.service
           ├─1029 /usr/sbin/gdm
           ├─1037 /usr/libexec/gdm-simple-slave --display-id
/org/gno...
           └─1047 /usr/bin/Xorg :0 -background none -verbose
-auth /r...

Oct 17 17:31:23 localhost systemd[1]: Started GNOME Display
Manager.
```

Пример: просмотр служб, отсортированных согласно порядку «запускать до запуска указанной службы»

Чтобы определить, какие службы настроены на запуск раньше запуска данной службы, выполните:

```
# systemctl list-dependencies --after gdm.service

gdm.service
├─dbus.socket
├─getty@tty1.service
├─livesys.service
├─plymouth-quit.service
├─system.slice
├─systemd-journald.socket
├─systemd-user-sessions.service
└─basic.target
[output truncated]
```

Пример: просмотр служб, отсортированных согласно порядку «запускать после запуска указанной службы»

Чтобы определить, какие службы настроены на запуск после запуска данной службы, выполните:

```
# systemctl list-dependencies --before gdm.service
gdm.service
├─dracut-shutdown.service
├─graphical.target
│ ├─systemd-readahead-done.service
│ ├─systemd-readahead-done.timer
│ └─systemd-update-utmp-runlevel.service
└─shutdown.target
```

```
├─systemd-reboot.service
└─final.target
  └─systemd-reboot.service
```

14.4.5. Запуск службы

Чтобы запустить юнит службы, соответствующий системной службе, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl start <имя>.service
```

Замените <имя> именем юнита службы, которую нужно запустить (например, gdm). Данная команда запускает выбранный юнит службы в текущей сессии. Сведения о том, как включить юнит службы в автостарт при загрузке системы, см. в п. 14.4.8. «Включение службы». Сведения о том, как определить статус конкретного юнита службы, см. в п. 14.4.4. «Просмотр статуса службы».

Пример: запуск службы

Юнит службы для сервера Apache HTTP называется httpd.service. Чтобы включить этот юнит и запустить демон httpd в текущем сеансе, выполните:

```
# systemctl start httpd.service
```

14.4.6. Остановка службы

Чтобы остановить юнит, соответствующий системной службе, выполните:

```
# systemctl stop <имя>.service
```

Замените <имя> именем юнита службы, которую нужно остановить (например, bluetooth). Данная команда останавливает выбранный юнит службы в текущем сеансе. Информацию о том, как отключить юнит службы и предотвратить его запуск во время загрузки системы, см. в п. 14.4.9. «Отключение службы». Информацию о том, как определить статус определенного юнита службы, см. в п. 14.4.4. «Просмотр статуса службы».

Пример: остановка службы

Юнит службы для демона bluetoothd называется bluetooth.service. Чтобы отключить юнит службы и остановить демон bluetoothd в текущем сеансе, выполните:

```
# systemctl stop bluetooth.service
```

14.4.7. Перезапуск службы

Чтобы перезапустить юнит службы, соответствующий системной службе, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl restart <имя>.service
```

Замените <имя> именем юнита службы, которую нужно перезапустить (например, httpd). Данная команда останавливает выбранный юнит службы в текущем сеансе и немедленно запускает его снова.

Примечание. Если выбранный юнит не работал в момент выполнения команды, он также будет запущен. Чтобы systemd перезапускал юнит службы только в том случае, если соответствующая служба уже запущена, выполните следующую команду:

```
# systemctl try-restart <имя>.service
```

Некоторые системные службы дают возможность перезагрузить их параметры, не прерывая выполнения самой службы. Для этого служит следующая команда:

```
# systemctl reload <имя>.service
```

Обратите внимание, что системные службы, не поддерживающие эту возможность, полностью игнорируют данную команду. Для удобства команда `systemctl` также поддерживает команды типа «перезагрузить-или-перезапустить» и «перезагрузить-или-попробовать перезапустить». Сведения о том, как определить статус конкретного юнита службы, см. в п. 14.4.4. «Просмотр статуса службы».

Пример: перезапуск службы

Чтобы не допустить получение пользователями ненужных сообщений об ошибках или не полностью показанных веб-страниц, сервер Apache HTTP дает возможность изменять и перезагружать свои параметры без перезапуска службы и прерывания активно обрабатываемых запросов. Для этого выполните:

```
# systemctl reload httpd.service
```

14.4.8. Включение службы

Чтобы настроить автоматический запуск юнита службы при загрузке системы, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl enable <имя>.service
```

Замените `<имя>` именем юнита службы, который нужно включить (например, httpd). Данная команда читает раздел `[Install]` указанного юнита службы и создает соответствующие символьные ссылки на файл /usr/lib/systemd/system/<имя>.service в каталоге /etc/systemd/system/ и его подкаталогах. Тем не менее, эта команда не перезаписывает уже существующие ссылки. Чтобы удостовериться в том, что символьные ссылки создаются заново, выполните:

```
# systemctl reenable <имя>.service
```

Данная команда отключает указанный юнит службы и немедленно включает его снова. Сведения о том, как определить, запускается ли определенный юнит службы при загрузке системы, см. в п. 14.4.4. «Просмотр статуса службы». Сведения о том, как запустить службу в текущем сеансе, см. в п. 14.4.5. «Запуск службы».

Пример: включение службы

Чтобы настроить автоматический запуск сервера Apache HTTP при загрузке системы, выполните следующую команду:

```
# systemctl enable httpd.service

Created symlink from /etc/systemd/system/multi-
user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
```

14.4.9. Отключение службы

Чтобы предотвратить автоматический запуск юнита службы при загрузке системы, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl disable <имя>.service
```

Замените <имя> именем службы, которую нужно отключить (например, bluetooth). Данная команда читает раздел [Install] файла указанного юнита службы и удаляет соответствующие символьные ссылки на файл /usr/lib/systemd/system/<имя>.service из каталога /etc/systemd/system/ и его подкаталогов. Кроме того, любой юнит службы можно замаскировать для предотвращения его запуска вручную или другой службой. Для этого выполните:

```
# systemctl mask <имя>.service
```

Данная команда заменит файл /etc/systemd/system/<имя>.service символьной ссылкой на /dev/null, делая исходный файл юнита недоступным для systemd. Чтобы откатить это действие и убрать маску с юнита службы, выполните:

```
# systemctl unmask <имя>.service
```

Сведения о том, как определить, запускается ли определенный юнит службы при загрузке системы, см. в п. 14.4.4. «Просмотр статуса службы». Сведения о том, как остановить работу службы в текущем сеансе, см. в п. 14.4.6. «Остановка службы».

Пример: отключение службы

Пример «Остановка службы» показывает, как остановить юнит bluetooth.service в текущем сеансе. Чтобы предотвратить запуск этого юнита службы во время загрузки системы, выполните:

```
# systemctl disable bluetooth.service
Removed symlink
/etc/systemd/system/bluetooth.target.wants/bluetooth.service.
Removed symlink /etc/systemd/system/dbus-org.bluez.service.
```

14.4.10. Запуск конфликтующей службы

В systemd существуют положительные и отрицательные зависимости между службами. Для запуска какой-либо службы может понадобиться запуск (положительная зависимость) или остановка (отрицательная зависимость) других служб.

При попытке запуска новой службы systemd автоматически разрешает все зависимости. Обратите внимание, что что делается без явного уведомления пользователя. Если служба уже запущена, и производится попытка запустить другую службу в отрицательной зависимостью, первая служба автоматически останавливается.

Если, например, в системе работает служба postfix, и будет выполнена попытка запустить службу sendmail, systemd сначала автоматически остановит postfix, т. к. эти две службы конфликтуют между собой и не могут работать на одном и том же порте одновременно.

14.4.11. Работа с целями systemd

В ранних версиях ОС РОСА, распространяемых с SysV init или Upstart, был реализован предварительно настроенный набор уровней выполнения (runlevel), представляющих собой определенные режимы работы системы. Эти уровни выполнения были пронумерованы от 0 до 6 и определялись набором системных служб, которые должны были запускаться, когда системный администратор активировал какой-то уровень выполнения.

В ОС РОСА «КОБАЛЬТ» понятие уровней выполнения было заменено на **цели systemd**.

Цели systemd представлены юнитами цели. Юниты цели имеют расширение .target, и их единственным назначением является группирование других юнитов systemd с помощью цепочек зависимостей. Например, юнит graphical.target, используемый для запуска графического сеанса работы, запускает системные службы, такие как менеджер графического входа в систему GNOME Display Manager (gdm.service) или служба учетных записей (accounts-daemon.service), а также включает юнит multi-user.target. Далее юнит multi-user.target запускает другие важные системные службы, такие как NetworkManager (NetworkManager.service) или D-Bus (dbus.service), и включает другой юнит цели, который называется basic.target.

В состав ОС РОСА «КОБАЛЬТ» входит несколько предварительно настроенных целей, которые более или менее аналогичны стандартному набору уровней выполнения, используемых в предыдущих релизах ОС. Из соображений совместимости также предоставляются псевдонимы для этих целей, которые напрямую отображают цели на уровни выполнения SysV. В таблице 6 приведен полный список уровней выполнения SysV и соответствующих им целей systemd.

Таблица 6. Сопоставление уровней выполнения SysV и целей systemd

Уровень выполнения	Юниты целей	Описание
0	runlevel0.target, poweroff.target	Завершение работы и выключение ОС
1	runlevel1.target, rescue.target	Аварийный командный интерпретатор
2	runlevel2.target, multi-user.target	Многопользовательская ОС без графического режима
3	runlevel3.target, multi-user.target	Многопользовательская ОС без графического режима
4	runlevel4.target, multi-user.target	Многопользовательская ОС без графического режима
5	runlevel5.target, graphical.target	Многопользовательская ОС с графическим режимом
6	runlevel6.target, reboot.target	Выключение и перезагрузка ОС

Чтобы просмотреть, изменить или настроить цели systemd, используйте утилиту *systemctl* так, как это описано в таблице 7 и в пунктах ниже. Команды `runlevel` и `telinit` по-прежнему доступны в ОС и работают, как ожидается, но включены только из соображений совместимости, и их использования следует избегать.

Таблица 7. Сопоставление команд SysV init и systemctl

Старая ко-	Новая команда	Описание

манда		
runlevel	systemctl list-units --type target	Выводит список юнитов цели, загруженных в данный момент.
telinit runlevel	systemctl isolate <имя>.target	Изменяет текущую цель.

14.4.12. Просмотр цели по умолчанию

Чтобы узнать, какая из целей используется по умолчанию, выполните следующую команду:

```
$ systemctl get-default
```

Данная команда разрешает символьную ссылку /etc/systemd/system/default.target и показывает результат. Сведения о том, как изменить цель по молчанию, см. в п. 14.4.14. «Смена цели по умолчанию». Сведения о том, как получить список всех загруженных на данный момент целей, см. в п. 14.4.13. «Просмотр текущей цели».

Пример: просмотр цели по умолчанию

Чтобы просмотреть юнит цели по умолчанию, выполните следующую команду:

```
$ systemctl get-default
graphical.target
```

14.4.13. Просмотр текущей цели

Чтобы получить список всех юнитов цели, загруженных на данный момент, выполните следующую команду:

```
$ systemctl list-units --type target
```

Для каждого юнита цели данная команда показывает его полное имя (`UNIT`), примечание, был ли юнит загружен (`LOAD`), статус его высокоуровневой (`ACTIVE`) и низкоуровневой (`SUB`) активаций, а также короткое описание (`DESCRIPTION`). По умолчанию команда `systemctl list-units` показывает только активные юниты. Чтобы просмотреть список всех загруженных юнитов вне зависимости от их статуса, выполните эту команду с ключом `--all` или `-a`:

```
systemctl list-units --type target --all
```

Сведения о том, как узнать цель по умолчанию, см. в п. 14.4.12. «Просмотр цели по умолчанию». Сведения о том, как сменить текущую цель, см. в п. 14.4.14. «Смена цели по умолчанию».

Пример: просмотр цели по умолчанию

Чтоб получить список всех юнитов, загруженных на данный момент, выполните:

```
$ systemctl list-units --type target

UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
```

```
graphical.target      loaded active active Graphical Interface
local-fs-pre.target   loaded active active Local File Systems
(Pre)
local-fs.target       loaded active active Local File Systems
multi-user.target     loaded active active Multi-User System
network.target        loaded active active Network
paths.target          loaded active active Paths
remote-fs.target      loaded active active Remote File Systems
sockets.target        loaded active active Sockets
sound.target          loaded active active Sound Card
spice-vdagentd.target loaded active active Agent daemon for Spice
guests
swap.target           loaded active active Swap
sysinit.target        loaded active active System Initialization
time-sync.target      loaded active active System Time
Synchronized
timers.target         loaded active active Timers

LOAD   = Reflects whether the unit definition was properly
loaded.
ACTIVE = The high-level unit activation state, i.e.
generalization of SUB.
SUB    = The low-level unit activation state, values depend on
unit type.

17 loaded units listed. Pass --all to see loaded but inactive
units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

14.4.14. Смена цели по умолчанию

Чтобы настроить систему на использование другой цели по умолчанию, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl set-default <имя>.target
```

Замените <имя> именем юнита цели, который нужно использовать по умолчанию (например, multi-user). Данная команда заменяет файл /etc/systemd/system/default.target символьной ссылкой, указывающей на /usr/lib/systemd/system/<имя>.target, где <имя> — это имя юнита цели, который нужно использовать. Сведения о том, как сменить текущую цель, см. в п. 14.4.15. «Смена текущей цели». Сведения о том, как получить список всех загруженных на данный момент юнитов цели, см. в п. 14.4.13. «Просмотр текущей цели».

Пример: смена цели по умолчанию

Чтобы настроить систему на выполнение юнита multi-user.target по умолчанию, выполните:

```
# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
```

```
ln -s '/usr/lib/systemd/system/multi-user.target'
'/etc/systemd/system/default.target'
```

14.4.15. Смена текущей цели

Чтобы установить другую цель в текущем сеансе, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl isolate <имя>.target
```

Замените <имя> на имя юнита цели, который нужно использовать, например, multi-user. Данная команда запускает юнит цели с указанным именем со всеми юнитами-зависимостями и немедленно останавливает все остальные. Сведения о том, как сменить цель по умолчанию, см. в п. 14.4.14. «Смена цели по умолчанию». Сведения о том, как получить список всех загруженных на данный момент юнитов цели, см. в п. 14.4.13. «Просмотр текущей цели».

Пример: смена текущей цели

Чтобы отключить графический интерфейс пользователя и в текущем сеансе сменить текущую цель на юнит multi-user.target, выполните:

```
# systemctl isolate multi-user.target
```

14.4.16. Установка режима восстановления

Режим восстановления (rescue) предоставляет удобное однопользовательское окружение, в котором администратор имеет возможность исправить ошибки в системе, препятствующие ее нормальной загрузке. В режиме восстановления система пытается смонтировать все локальные файловые системы и запустить некоторые важные системные службы, но не активирует сетевые интерфейсы и не позволит другим пользователям выполнить одновременный вход. В ОС РОСА «КОБАЛЬТ» режим восстановления равноценен однопользовательскому режиму и требует пароля root.

Чтобы сменить текущую цель и войти в режим восстановления в текущем сеансе, выполните следующую команду с привилегиями суперпользователя:

```
# systemctl rescue
```

Данная команда аналогична команде `systemctl isolate rescue.target`, но помимо прочего также посылает информационное сообщение всем пользователям, выполнившим на данный момент вход в систему. Чтобы systemd не посылал этого сообщения, выполните данную команду с ключом `--no-wall`:

```
systemctl --no-wall rescue
```

Пример: вход в режим восстановления

```
# systemctl rescue


Broadcast message from root@localhost on pts/0 (Fri 2013-10-25
18:23:15 CEST):
The system is going down to rescue mode NOW!
```

14.4.17. Установка аварийного режима

Аварийный режим (emergency) предоставляет самое минимальное окружение из

всех возможных, которое позволяет исправить ошибки системы даже в тех ситуациях, когда она не в состоянии войти в режим восстановления. В аварийном режиме ОС монтирует корневую файловую систему только для чтения, не пытается смонтировать никаких других локальных ФС, не активирует сетевые интерфейсы и запускает только несколько самых важных служб. В ОС РОСА «КОБАЛЬТ» аварийный режим требует пароля root.

Чтобы сменить текущую цель и войти в аварийный режим, выполните следующую команду с привилегиями суперпользователя:

```
# systemctl emergency
```

Данная команда аналогична команде `systemctl isolate emergency.target`, но помимо прочего также посылает информационное сообщение всем пользователям, выполнившим на данный момент вход в систему. Чтобы systemd не посылал этого сообщения, выполните данную команду с ключом `--no-wall`:

```
# systemctl --no-wall emergency
```

14.4.18. Выключение системы, режим ожидания и спящий режим

В ОС РОСА «КОБАЛЬТ» утилита systemctl заменила несколько команд, управляющих питанием, которые использовались в предыдущих версиях системы. Команды, список которых приведен в таблице 8, по-прежнему доступны в системе из соображений совместимости, но systemctl рекомендуется использовать всегда, когда это возможно.

Таблица 8. Сопоставление команд управления питанием с systemctl

Старая команда	Новая команда	Описание
`halt`	`systemctl halt`	Останавливает систему
`poweroff`	`systemctl poweroff`	Выключает систему
`reboot`	`systemctl reboot`	Перезагружает систему
`pm-suspend`	`systemctl suspend`	Переводит систему в режим ожидания
`pm-hibernate`	`systemctl hibernate`	Переводит систему в спящий режим
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Переводит систему в гибридный спящий режим

14.4.19. Выключение системы

Утилита systemctl предоставляет команды для выключения системы, но традиционная команда shutdown также поддерживается. Хотя для выполнения действий по выключению команда shutdown вызывает утилиту systemctl, ее преимущество в том, что она также поддерживает и аргумент времени. Это особенно удобно для запланированного обслуживания, а также предоставляет пользователям время для реакции на предупреждение о том, что было запланировано выключение системы. Возможность отменить выключение системы также может быть преимуществом.

14.4.19.1. Использование команды systemctl

Чтобы завершить работу ОС и выключить питание компьютера, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl poweroff
```

Чтобы завершить работу ОС без отключения питания, выполните:

```
# systemctl halt
```

По умолчанию при запуске одной из этих команд systemd посылает информационное сообщение всем пользователям, на данный момент выполнившим вход в систему. Чтобы systemd не посылал этого сообщения, выполните указанную команду с ключом `--no-wall`, например:

```
systemctl --no-wall poweroff
```

14.4.19.2. Использование команды shutdown

Чтобы выключить систему и обесточить машину в определенное время, выполните следующую команду с привилегиями суперпользователя root:

```
# shutdown --poweroff чч:мм
```

Здесь `чч:мм` — это время в 24-часовом формате. За 5 минут до выключения системы создается файл /run/nologin, запрещающий пользователям вход в систему. При использовании аргумента времени к команде можно добавить необязательное сообщение `wall`.

Чтобы выключить и остановить систему после некоторой задержки без отключения питания компьютера, выполните:

```
# shutdown --halt +<m>
```

Здесь `+<m>` — время задержки в минутах. Ключевое слово `now` является псевдонимом для `+0`.

Пользователь root может отменить ожидаемое выключение следующим образом:

```
# shutdown -c
```

Дополнительные возможности и параметры см. на странице руководства shutdown(8).

14.4.20. Перезагрузка системы

Чтобы перезагрузить систему, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl reboot
```

При запуске этой команды по умолчанию systemd посылает информационное сообщение всем пользователям, на данный момент выполнившим вход в систему. Чтобы systemd не посылал этого сообщения, выполните указанную команду с ключом `--no-wall`, например:

```
systemctl --no-wall reboot
```

14.4.21. Перевод системы в режим ожидания

Чтобы перевести систему в режим ожидания, выполните следующую команду с

привилегиями суперпользователя root:

```
# systemctl suspend
```

Данная команда сохраняет состояние системы в оперативной памяти, и отключает питание большинства устройств компьютера, за исключениям модулей памяти. Далее при возобновлении работы компьютера система восстанавливает свое состояние из оперативной памяти без необходимости в полной загрузке. Поскольку состояние системы сохраняется в оперативной памяти, а не на жестком диске, восстановление системы из режима ожидания происходит значительно быстрее, чем из спящего режима. Однако компьютер в режиме ожидания является уязвимым к перебоям питания.

14.4.22. Перевод системы в спящий режим

Чтобы перевести систему в спящий режим, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl hibernate
```

Данная команда сохраняет состояние системы на жесткий диск и выключает питание. Далее при повторном включении компьютера система во восстанавливает свое состояние из сохраненных данных без необходимости в полной загрузке. Поскольку состояние системы сохраняется на жестком диске, а не в оперативной памяти, отпадает необходимость поддерживать электропитание для модулей ОЗУ, но, соответственно, процесс восстановления системы из спящего режима происходит значительно медленнее, чем из режима ожидания. Чтобы перевести систему в гибридный спящий режим, выполните:

```
# systemctl hybrid-sleep
```

14.4.23. Управление systemd на удаленной машине

Помимо локального управления systemd и service manager, утилита systemctl также предоставляет возможность взаимодействия с systemd на удаленной машине с использованием протокола SSH. При условии, что на удаленной машине служба выполняется sshd, подключиться к этой машине можно, выполнив команду `systemctl` с ключом `--host` или `-H`:

```
systemctl --host <имя_пользователя>@<имя_хоста> <команда>
```

Замените `<имя_пользователя>` именем удаленного пользователя, `<имя_хоста>` — именем хоста машины, а `<команду>` — одной из команд systemctl, описанных выше. Обратите внимание, что для того, чтобы указанный пользователь смог получить удаленный доступ с использованием протокола SHH, удаленная машина должна быть настроена так, чтобы разрешить ему сделать это.

Пример: удаленное управление

Чтобы выполнить вход на удаленной машине с именем server-01.example.com с привилегиями суперпользователя root и определить текущий статус юнита httpd.service, выполните следующую команду:

```
# systemctl -H root@server-01.example.com status httpd.service
```

```
>>>>>>> systemd unit files -- update
```

```
root@server-01.example.com's password:
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service;
enabled)
   Active: active (running) since Fri 2013-11-01 13:58:56 CET; 2h
48min ago
 Main PID: 649
   Status: "Total requests: 0; Current requests/sec: 0; Current
traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
```

14.4.24. Создание и изменение файлов юнитов systemd

Файл юнита содержит конфигурационные директивы, описывающие юнит и определяющие его поведение. Несколько команд systemctl работают с файлами юнита в фоновом режиме. Чтобы более тонко отрегулировать поведение юнита, системный администратор должен отредактировать файл вручную. В таблице «Местоположение файлов юнитов systemd» приводятся три основных каталога, в которых располагаются файлы юнитов. Каталог /etc/systemd/system/ зарезервирован для файлов юнитов, созданных или измененных системным администратором.

Имена файлов юнитов имеют формат «<имя_юнита>.<тип_расширение>». Полный список типов юнитов см. в таблице 1. В системе обычно присутствуют юниты sshd.service и sshd.socket.

Файлы юнитов могут быть дополнены каталогом для дополнительных конфигурационных файлов. Чтобы, например, добавить пользовательские параметры в sshd.service, создайте файл sshd.service.d/custom.conf и поместите в него дополнительные директивы. Более подробные сведения о каталогах конфигурационных файлов см. в п. «Изменение существующих файлов юнитов» на стр. 180.

Также можно создать каталоги sshd.service.wants/ и sshd.service.requires/, содержащие символьные ссылки на файлы юнитов, которые являются зависимостями службы sshd. Эти символьные ссылки автоматически создаются либо во время процесса установки согласно параметрам для файлов юнитов, указанным в разделе [Install] (см. таблицу 11), либо в процессе работы согласно параметрам раздела [Unit] (см. таблицу 9). Также можно создать эти каталоги и символьные ссылки вручную.

Многие параметры файлов юнитов можно настроить с помощью так называемых спецификаторов юнитов — записей с символами замены, динамически заменяемых параметрами юнитов при загрузке файлов юнитов. Это дает возможность создавать универсальные файлы юнитов, служащие шаблонами для создаваемых экземпляров.

14.4.24.1. Структура файла юнита

Файл юнита обычно состоит из трех разделов:

1) `[Unit]` — содержит общие параметры, не зависящие от типа юнита. Эти параметры предоставляют описание юнита, определяют его поведение и настраивают зависимости для других юнитов. Список наиболее часто используемых параметров `[Unit]` см. в таблице 9.

2) `[unit type]` — если у юнита есть директивы, характерные для данного типа юнита, они собраны в разделе, названном по типу юнита. Файлы юнитов служб, например, содержат раздел `[Service]`. Список наиболее часто используемых параметров `[Service]` см. в таблице 10.

3) `[Install]` — содержит информацию об установке юнитов, которая используется командами `systemctl enable` и `disable`. Список параметров `[Install]` см. в таблице 11.

Таблица 9. Важные параметры раздела [Unit]

Параметр	Описание
`Description`	Значимое описание юнита. Этот текст показывается, например, в выводе команды `systemctl status`
`Documentation`	Список адресов URI, ссылающихся на документацию для данного юнита
`After`	Определяет порядок, в котором запускаются юниты. Юнит начинает работу только после того, как становятся активными юниты, указанные параметром `After`. В отличие от параметра `Requires`, параметр `After` не выполняет явную активацию указанных юнитов. Параметр `Before` имеет действие, противоположное параметру `After`
`Requires`	Настраивает зависимости от других юнитов. Юниты, перечисленные в параметре `Requires`, активируются вместе с изначальным юнитом. Если запуск какого-то из перечисленных юнитов окончится неудачей, изначальный юнит не будет активирован
`Wants`	Настраивает более слабые зависимости, чем `Requires`. Если запуск какого-то из перечисленных юнитов окончится неудачей, это не повлияет на запуск исходного юнита. Это рекомендованный способ настройки пользовательских зависимостей юнитов
`Conflicts`	Настраивает отрицательные зависимости, в противоположность `Requires`

Примечания.

1. Полный список параметров, настраиваемых в разделе `[Unit]`, см. на странице руководства systemd.unit(5).

2. В большинстве случаев достаточно настроить порядковые зависимости параметров `After` и `Before`. При настройке также и требуемых зависимостей при помощи параметра `Wants` (рекомендуемый способ) или `Requires` зависимости порядка все равно нужно указывать. Это происходит потому, что зависимости порядка и требуемые зависимости действуют независимо друг от друга.

Таблица 10. Важные параметры раздела [Service]

Параметр	Описание

`Type`	Настраивает тип запуска процесса юнита, влияющий на функционал `ExecStart` и связанные с ним параметры. Может быть одним из: • `simple` — значение по умолчанию. Процесс, запущенный с помощью ExecStart, является главным процессом службы; • `forking` — процесс, запущенный с помощью ExecStart, порождает другой процесс, который становится главным процессом службы. После завершения запуска родительский процесс завершается; • `oneshot` — этот тип аналогичен типу `simple`, но процесс завершается до запуска последующих юнитов; • `dbus` — этот тип аналогичен типу `simple`, но последующие юниты запускаются только после того, как главный процесс получает имя D-Bus; • `notify` — этот тип аналогичен типу `simple`, но последующие юниты запускаются только после того, как будет послано уведомительное сообщение с помощью функции sd_notify(); • `idle` — аналогичен `simple`. Фактическое выполнение бинарного файла службы откладывается до окончания выполнения всех задач, что помогает избежать смешения вывода сообщений статуса с выводом сообщений служб из командного интерпретатора
`ExecStart`	Указывает команды или сценарии, которые должны выполняться при запуске юнита. `ExecStartPre` и `ExecStartPost` указывают на пользовательские команды, которые должны запуститься до и после `ExecStart`. `Type=oneshot` включает возможность указать несколько пользовательских команд, которые затем выполняются последовательно
`ExecStop`	Указывает команды или сценарии, которые должны выполняться при остановке юнита
`ExecReload`	Указывает команды или сценарии, которые должны выполняться при перезагрузке юнита
`Restart`	Если этот параметр включен, после завершения ее процесса служба перезапускается, за исключением «чистой остановки» (clean stop), выполненной с помощью команды systemctl
`RemainAfterExit`	Если значение установлено как истинное, служба считается активной даже после завершения всех ее процессов. Значение по умолчанию — неверно. Этот параметр особенно полезен при настроенном параметре `Type=oneshot`

Примечание. Полный список параметров, настраиваемых в разделе `[Service]`, см. на странице руководства systemd.service(5).

Таблица 11. Важные параметры раздела [Install]

Параметр	Описание
`Alias`	Предоставляет список дополнительных имен юнита, разделенных пробелами. Большинство команд systemctl, исключая `systemctl enable`, вместо фактических имен юнитов могут использовать псевдонимы
`RequiredBy`	Список юнитов, зависящих от данного юнита. При включении данного юнита юниты, перечисленные в `RequiredBy`, получают зависимость `Require` относительно данного юнита
`WantedBy`	Список юнитов со слабой зависимостью от данного юнита. При включении данного юнита юниты, перечисленные в `WantedBy`, получают зависимость Want относительно данного юнита
`Also`	Указывает список юнитов, которые должны быть установлены или удалены вместе с данным юнитом
`DefaultInstance`	Этот параметр ограничен создаваемыми экземплярами юнитов и указывает экземпляр по умолчанию, для которого включается юнит

Примечание. Полный список параметров, настраиваемых в разделе `[Install]`, см. на странице руководства systemd.unit(5).

Существует целый ряд параметров, которые можно использовать для более точной настройки конфигурации юнита. Пример «Файл юнита postfix.service» показывает образец юнита службы, установленного в системе. Кроме того, параметры файла юнита можно настроить так, чтобы активировать динамическое создание юнитов.

Пример: файл юнита postfix.service

Далее приведено содержимое файла юнита /usr/lib/systemd/system/postifix.service в том виде, в котором он поставляется в пакете postfix:

```
[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop
```

```
[Install]
WantedBy=multi-user.target
```

В разделе `[Unit]` описывается служба, настраиваются порядковые зависимости и конфликтующие юниты. В разделе `[Service]` указывается последовательность сценариев, которые должны выполняться во время активации, остановки или перезагрузки юнита. Параметр `EnvironmentFile` указывает на местоположение определяемых переменных окружения службы, `PIDFile` — определяет стабильный PID главного процесса службы, и, наконец, в разделе `[Install]` перечисляются юниты, зависящие от службы.

14.4.24.2. Создание пользовательских файлов юнитов

Создать файлы юнитов с нуля можно несколькими способами: запустить пользовательский демон, создать второй экземпляр уже существующей службы (как показано в примере «Создание второго экземпляра службы sshd») или импортировать сценарий SysV init (подробности в п. «Преобразование сценариев инициализации SysV Init в файлы юнитов» на стр. 176). С другой стороны, если необходимо только изменить или расширить поведение существующего юнита, используйте инструкции из п. «Изменение существующих файлов юнитов» на стр. 180. Далее описывается общий процесс создания пользовательской службы.

1) Подготовьте запускаемый файл для пользовательской службы. Это может быть пользовательский сценарий или исполняемый файл, предоставленный поставщиком ПО. При необходимости подготовьте файл PID для сохранения постоянного PID главного процесса пользовательской службы. Также можно добавить переменные командного интерпретатора в файлы окружения. Убедитесь в том, что исходный сценарий доступен для выполнения (`chmod a+x`) и не является интерактивным.
2) Создайте файл юнита в каталоге /etc/systemd/system/ и убедитесь, что права доступа выставлены корректно. Выполните с привилегиями суперпользователя root следующие команды:

```
# touch /etc/systemd/system/<имя>.service
# chmod 664 /etc/systemd/system/<имя>.service
```

Замените `<имя>` именем создаваемой службы. Обратите внимание, что файл не обязательно должен быть исполняемым.

3) Откройте файл <имя>.service, созданный на предыдущем шаге, и добавьте параметры настройки службы. В зависимости от типа используемой службы существует значительное разнообразие параметров. См. п. 14.4.24.1. «Структура файла юнита». Ниже приведен пример конфигурации юнита для сетевой службы:

```
[Unit]
Description=<описание_службы>
After=network.target

[Service]
ExecStart=<путь_до_исполняемого_файла>
Type=forking
PIDFile=<путь_до_файла_PID>
```

```
[Install]
WantedBy=default.target
```

Здесь:

- `<описание_службы>` — информационное описание, показываемое в файлах журналов и в выводе команды `systemctl status`;
- параметр `After` обеспечивает запуск службы только после запуска юнита network. Добавьте список других релевантных служб или целей, разделенных пробелами;
- `<путь_до_исполняемого_файла>` — путь до фактического исполняемого файла службы;
- `Type=forking` используется для демонов, делающих системный вызов fork. Главный процесс службы создается с PID, указанным в значении `<путь_до_файла_PID>`. Другие типы запуска ищите в таблице 10;
- `WantedBy` указывает цель или цели, в которых должна запускаться служба. Понимайте цели как замену старой концепции уровней выполнения (runlevels).

4) Уведомите систему о существовании нового файла <имя>.service, выполнив следующие команды:

```
# systemctl daemon-reload
# systemctl start <имя>.service
```

Примечание. После создания новых файлов юнитов или изменения существующих всегда запускайте команду `systemctl daemon-reload`. В противном случае команды `systemctl start` или `systemctl enable` могут закончиться неудачей из-за несоответствия между статусами systemd и фактическими файлами юнита службы на диске.

Теперь с юнитом <имя>.service можно работать как с любой другой системной службой согласно п. 14.4. «Управление системными службами».

Пример: создание файла emacs.service

При использовании текстового редактора Emacs часто бывает удобнее и быстрее запускать его в фоновом режиме, чем запускать новый экземпляр редактора каждый раз, когда нужно отредактировать файл. Далее описываются шаги, необходимые для создания файла юнита для Emacs, чтобы им можно было управлять, как службой.

1) Создайте файл юнита в каталоге /etc/systemd/system/ и убедитесь, что права доступа выставлены правильно. Выполните с привилегиями суперпользователя root:

```
# touch /etc/systemd/system/emacs.service
# chmod 664 /etc/systemd/system/emacs.service
```

2) Добавьте в файл следующее содержимое:

```
[Unit]
Description=Emacs: the extensible, self-documenting text editor

[Service]
Type=forking
ExecStart=/usr/bin/emacs --daemon
```

```
ExecStop=/usr/bin/emacsclient --eval "(kill-emacs)"
Environment=SSH_AUTH_SOCK=%t/keyring/ssh
Restart=always

[Install]
WantedBy=default.target
```

При использовании вышеприведенной конфигурации исполняемый файл /usr/bin/emacs запускается в режиме демона при запуске службы. Переменная окружения SSH_AUTH_SOCK настраивается с использованием указателя юнита `%t`", означающего каталог выполнения. Служба также перезапускает процесс emacs в случае его неожиданного завершения.

3) Для перезагрузки конфигурации и перезапуска службы выполните следующие команды:

```
# systemctl daemon-reload
# systemctl start emacs.service
```

Поскольку редактор теперь зарегистрирован в качестве системной службы, для него можно использовать все стандартные команды `systemctl`. Чтобы, например, просмотреть статус редактора, выполните `systemctl status emacs`, а чтобы настроить автоматический запуск редактора при загрузке системы, выполните `systemctl enable emacs`.

Пример: создание второго экземпляра службы sshd

Системным администраторам часто бывает необходимо настроить и запускать несколько экземпляров одной службы. Для этого копируются файлы конфигурации исходной службы, в которых изменяются некоторые параметры для избежания конфликтов с изначальным экземпляром службы. Ниже показывается, как создать второй экземпляр службы sshd.

1) Создайте копию файла sshd_config, которая будет использоваться вторым демоном:

```
# cp /etc/ssh/sshd{,-second}_config
```

2) Отредактируйте файл sshd-second_config, созданный на предыдущем шаге, чтобы присвоить второму демону другой номер порта и файл PID:

```
Port 22220
PidFile /var/run/sshd-second.pid
```

Подробности о параметрах Port и PidFile см. на странице руководства sshd_config(5). Убедитесь в том, что выбранный порт не используется какой-либо другой службой. Файл PID не должен существовать до запуска службы; он создается автоматически при ее запуске.

3) Создайте копию файла юнита systemd для службы sshd:

```
# cp /usr/lib/systemd/system/sshd.service
/etc/systemd/system/sshd-second.service
```

4) Измените файл sshd-second.service, созданный на предыдущем шаге, следующим образом:

а) отредактируйте параметр Description:

```
Description=OpenSSH server second instance daemon
```

б) добавьте службу sshd.service к службам, указанным в параметре `After`, так, чтобы второй экземпляр службы запускался только после запуска первого:

```
After=syslog.target network.target auditd.service sshd.service
```

в) удалите запись `ExecStartPre=/usr/sbin/sshd-keygen`, поскольку первый экземпляр sshd уже предусматривает создание ключа;

г) добавьте ключ `-f /etc/ssh/sshd-second_config` команде `sshd`, чтобы использовался альтернативный файл конфигурации:

```
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd-second_config
$OPTIONS
```

д) после внесения вышеуказанных изменений файл sshd-second.service должен выглядеть следующим образом:

```
[Unit]
Description=OpenSSH server second instance daemon
After=syslog.target network.target auditd.service sshd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd-second_config
$OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

5) Если используется SELinux, добавьте порт второго экземпляра sshd к портам SSH. В противном случае второму экземпляру sshd будет отказано в привязке к порту:

```
# semanage port -a -t ssh_port_t -p tcp 22220
```

6) Включите автоматический запуск службы sshd-second.service при загрузке системы:

```
# systemctl enable sshd-second.service
```

7) Проверьте с помощью команды `systemctl status`, выполняется ли sshd-second.service. Кроме того, проверьте доступность порта, подключившись к службе:

```
$ ssh -p 22220 user@server
```

Если используется межсетевой экран, обеспечьте его правильную конфигурацию для разрешения подключений ко второму экземпляру sshd.

Чтобы узнать, как правильно настроить очередь запуска цели и зависимости для пользовательских файлов юнитов, см. п. «Как написать файл юнита службы с принудительным запуском конкретных служб» на стр. 187 и п. «Как решить, какие зависимости должно иметь определение юнита службы systemd» на стр. 189.

Чтобы настроить лимиты для служб, запускаемых systemd, обратитесь к п. «Настройка лимитов для служб в ОС РОСА «КОБАЛЬТ» и systemd» на стр. 194. Эти лимиты настраиваются в файле юнита службы. Обратите внимание, что systemd игнорирует лимиты в файлах /etc/security/limits.conf и /etc/security/limits.d/*.conf. Лимиты, указанные в этих файлах, настраиваются PAM при запуске сеанса регистрации, но демоны, запускаемые systemd, не работают с сеансами регистрации PAM.

Преобразование сценариев инициализации SysV Init в файлы юнитов

Для преобразования сценария в файл юнита требуется проанализировать скрипт и выделить из него всю необходимую информацию. На основе этих данных можно составить файл юнита, как описано в п. 14.4.24.2. «Создание пользовательских файлов юнитов». Поскольку сценарии инициализации могут сильно отличаться друг от друга в зависимости от типа службы, администратору может потребоваться применить больше параметров переноса, чем описано в данном разделе. Обратите внимание, что некоторые уровни адаптации, доступные при использовании сценариев инициализации, больше не поддерживаются юнитами systemd. См. подраздел 14.3. «Изменения совместимости».

Большая часть информации, необходимой для преобразования, содержится в заголовке сценария. Ниже приведен пример вводной части сценария инициализации, используемого для запуска службы postfix:

```
#!/bin/bash
#
# postfix      Postfix Mail Transfer Agent
#
# chkconfig: 2345 80 30
# description: Postfix is a Mail Transport Agent, which is the
program \
#              that moves mail from one machine to another.
# processname: master
# pidfile: /var/spool/postfix/pid/master.pid
# config: /etc/postfix/main.cf
# config: /etc/postfix/master.cf

### BEGIN INIT INFO
# Provides: postfix MTA
# Required-Start: $local_fs $network $remote_fs
# Required-Stop: $local_fs $network $remote_fs
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop postfix
# Description: Postfix is a Mail Transport Agent, which is the
program that
#              moves mail from one machine to another.
### END INIT INFO
```

В примере выше обязательными являются только строки, начинающиеся с

`# chkconfig` и `# description`, поэтому в других файлах init их можно не найти. Текст между записями `### BEGIN INIT INFO` и `### END INIT INFO` называется заголовком Linux Standard Base (LSB). В случае наличия заголовок LSB содержит директивы, содержащие описание службы, зависимости и уровни запуска по умолчанию. Далее следует обзор аналитических задач, которые направлены на сбор данных, необходимых для нового файла юнита. Сценарий инициализации приведен как пример; итоговый файл юнита postfix см. в примере «Файл юнита postfix.service».

Где найти описание службы

Информация, описывающая сценарий, находится в записи, начинающейся с `# description`. Используйте это описание вместе с именем службы в параметре `Description` раздела `[Unit]` в файле юнита. Заголовок LSB может содержать аналогичные данные в строках `# Short-Description` и `# Description`.

Где найти информацию о зависимостях службы

Заголовок LSB должен содержать несколько директив, образующих зависимости между службами. Большая часть из них переводится в параметры юнита systemd, как показано в таблице 12.

Таблица 12. Параметры зависимостей в заголовке LSB

Параметр LSB	Описание	Аналог в файле юнита
`Provides`	Указывает имя согласно средству загрузки службы, на которое можно ссылаться в других сценариях инициализации (с помощью префикса $). Больше не требуется, т. к. файлы юнитов ссылаются на другие юниты по имени их файлов	—
`Required-Start`	Содержит имена средств загрузки требуемых служб. Этот параметр преобразуется в порядковую зависимость. Имена загрузочных средств заменяются именами файлов юнитов соответствующих служб или целей, к которым они принадлежат. В случае postfix, например, зависимость `Required-Start` для $network преобразовывается в зависимость `After` в файле network.target	`After, Before`
`Should-Start`	Более слабые зависимости, чем `Required-Start`. Несоблюденные зависимости `Should-Start` не влияют на запуск службы	`After, Before`
`Required-Stop, Should-Stop`	Отрицательные зависимости	`Conflicts`

Как найти цели по умолчанию для службы

Строка, начинающаяся с `# chkconfig`, содержит три числовых значения. Самое важное из них — первое, представляющее уровни выполнения по умолчанию, в которых запускается служба. Чтобы сопоставить уровни выполнения с соответствующими целями systemd, используйте таблицу 6. Затем перечислите эти цели в параметре `WantedBy` раздела `[Install]` файла юнита. Например: раньше postfix запускался на уровнях выполнения 2, 3, 4 и 5, что соответствует целям multi-user.target и graphical.target в данной версии ОС РОСА «КОБАЛЬТ». Обратите внимание, что graphical.target зависит от multiuser.target, поэтому не обязательно указывать обе цели, как показано в примере «Файл юнита postfix.service». Найти информацию об уровне по умолчанию и о запрещенном уровне можно в строках `#Default-Start` и `#Default-Stop` заголовка LSB.

Другие два значения, присутствующие в строке `# chkconfig`, обозначают приоритеты запуска и выключения сценария инициализации. Эти значения обрабатываются systemd, если она загружает непосредственно сам сценарий, но в файле юнита эквивалентов им нет.

Как найти файлы, используемые службой

Для сценариев инициализации требуется загрузка библиотеки функций из соответствующего каталога, и есть возможность импорта файлов конфигурации, окружения и PID. Переменные окружения указываются в строке в заголовке сценария инициализации, начинающейся с `# config`, которая преобразуется в параметр файла юнита `EnvironmentFile`. Файл PID, указанный в строке `# pidfile` сценария инициализации, импортируется в файл юнита с помощью параметра `PIDFile`.

Основная информация, которая не включается в заголовок сценария инициализации, это путь до исполняемого файла службы, а также, возможно, до некоторых других файлов, требуемых для службы. В предыдущих версиях ОС РОСА в сценариях инициализации указывался оператор условия Bash, определяющий поведение службы при действиях по умолчанию, таких как запуск, остановка или перезапуск, а также действиях, определенных пользователем. В следующем фрагменте сценария инициализации postfix показан блок кода, который должен выполняться при запуске службы.

```
conf_check() {
    [ -x /usr/sbin/postfix ] || exit 5
    [ -d /etc/postfix ] || exit 6
    [ -d /var/spool/postfix ] || exit 5
}

make_aliasesdb() {
   if [ "$(/usr/sbin/postconf -h alias_database)" ==
"hash:/etc/aliases" ]
   then
       # /etc/aliases.db might be used by other MTA, make sure
nothing
       # has touched it since our last newaliases call
       [ /etc/aliases -nt /etc/aliases.db ] ||
```

```
            [ "$ALIASESDB_STAMP" -nt /etc/aliases.db ] ||
            [ "$ALIASESDB_STAMP" -ot /etc/aliases.db ] || return
        /usr/bin/newaliases
        touch -r /etc/aliases.db "$ALIASESDB_STAMP"
    else
        /usr/bin/newaliases
    fi
}

start() {
    [ "$EUID" != "0" ] && exit 4
    # Check that networking is up.
    [ ${NETWORKING} = "no" ] && exit 1
    conf_check
    # Start daemons.
    echo -n $"Starting postfix: "
    make_aliasesdb >/dev/null 2>&1
    [ -x $CHROOT_UPDATE ] && $CHROOT_UPDATE
    /usr/sbin/postfix start 2>/dev/null 1>&2 && success || failure
$"$prog start"
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch $lockfile
         echo
    return $RETVAL
}
```

Возможность расширения сценария инициализации разрешала указывать два пользовательских действия, `conf_check()` и `make_aliasesdb()`, которые вызывались из блока функции `start()`. При внимательном рассмотрении в коде выше упоминаются несколько внешних файлов и каталогов: основной исполняемый файл службы /usr/sbin/postfix, каталоги, содержащие файлы конфигурации /etc/postfix/ и /var/spool/postfix/, а также каталог /usr/sbin/postconf/.

Systemd поддерживает только предварительно настроенные действия, но параметры `ExecStart`, `ExecStartPre`, `ExecStartPost`, `ExecStop` и `ExecReload` дают возможность выполнения также и пользовательских файлов. В случае службы postfix в ОС РОСА «КОБАЛЬТ» при запуске службы файл /usr/sbin/postfix выполняется вместе со сценариями поддержки. Обратитесь к файлу юнита postfix в примере «Файл юнита postfix.service» на стр. 171.

Преобразование сложных сценариев инициализации требует понимания смысла каждого выражения в сценарии. Некоторые из них характерны только для конкретной версии ОС, поэтому их преобразовывать не нужно. С другой стороны, для соответствия новому окружению нужно внести некоторые изменения как в файлы юнитов, так и в файлы поддержки.

Изменение существующих файлов юнитов

Для служб, установленных в системе, существуют файлы юнитов по умолчанию в каталоге /usr/lib/systemd/system/. Системные администраторы не должны вносить изменения в эти файлы напрямую. Для этих целей существует каталог /etc/systemd/system/. В зависимости от масштаба требуемых изменений, предлагается один из следующих подходов:

1) Создание каталога для дополнительного файла конфигурации в /etc/systemd/system/unit.d/. Этот способ рекомендуется для большинства случаев и предполагает расширение конфигурации по умолчанию при обращении к исходному файлу юнита. Соответственно, изменения в юните по умолчанию, вносимые при обновлении пакета, применяются по умолчанию. Подробности см. в п. «Расширение исходной конфигурации юнита» на стр. 182.
2) Копирование исходного файла юнита из /usr/lib/systemd/system/ в /etc/systemd/system/ и последующее внесение изменений в файл копии. Копия перезаписывает исходный файл, соответственно, изменения в юните по умолчанию, вносимые при обновлении пакета, не применяются. Этот способ удобен, когда в юнит нужно внести многочисленные изменения, которые должны оставаться неизменными вне зависимости от обновления пакетов. См. п. «Изменение исходной конфигурации юнита» на стр. 183.

Чтобы вернуть исходную конфигурацию юнита, просто удалите созданные файлы из /etc/systemd/system/. Для применения изменений к файлам юнитов без перезагрузки системы выполните следующую команду:

```
systemctl daemon-reload
```

Параметр `daemon-reload` перезагружает все файлы юнитов и заново создает все дерево зависимостей, что необходимо для немедленного применения любого изменения в файле юнита. Как вариант, тот же самый результат можно получить с помощью следующей команды:

```
init q
```

Кроме того, если измененный файл юнита принадлежит к выполняемой службе, для применения новых параметров эта служба должна быть перезапущена:

```
systemctl restart <имя>.service
```

Примечание. Чтобы изменить зависимости или пределы срока ожидания службы, управляемой сценарием инициализации SysV, не изменяйте сам сценарий. Вместо этого создайте для службы дополнительный файл конфигурации systemd, как описано в п. «Расширение исходной конфигурации юнита» на стр. 182 и в п. «Изменение исходной конфигурации юнита» на стр. 183. После этого обращайтесь с этой службой точно так же, как и с обычной службой systemd.

Чтобы, например, расширить конфигурацию службы network, не изменяйте файл сценария инициализации /etc/rc.d/init.d/network. Вместо этого создайте новый каталог /etc/systemd/system/network.service.d/ и системный файл-дополнение /etc/systemd/system/network.service.d/my_config.conf. Затем сохраните измененные значения в файле-дополнении.

Примечание. systemd знает службу network под именем network.service, поэтому созданный каталог и нужно назвать network.service.d.

Расширение исходной конфигурации юнита

Чтобы расширить исходный файл юнита дополнительными возможностями, сначала нужно создать каталог в /etc/systemd/system/. При необходимости расширить юнит службы выполните следующую команду с привилегиями суперпользователя root:

```
# mkdir /etc/systemd/system/<имя>.service.d/
```

Замените <имя> именем службы, которую нужно расширить. Синтаксис, указанный выше, применяется ко всем типам юнитов.

Создайте конфигурационный файл в каталоге, созданном на предыдущем шаге. Обратите внимание, что имя файла должно содержать суффикс .conf. Выполните:

```
# touch /etc/systemd/system/name.service.d/config_<имя>.conf
```

Замените config_<имя> на имя конфигурационного файла. Структура этого файла следует структуре обычного файла юнита, поэтому все директивы должны указываться в соответствующих разделах, см. «Структура файлов юнитов».

Чтобы, например, добавить зависимость, создайте конфигурационный файл со следующим содержимым:

```
[Unit]
Requires=<новая_зависимость>
After=<новая_зависимость>
```

Здесь <новая_зависимость> означает юнит, который будет помечен, как зависимость. Другой пример — конфигурационный файл, перезапускающий службу после окончания работы ее главного процесса с задержкой 30 секунд:

```
[Service]
Restart=always
RestartSec=30
```

Рекомендуется создавать небольшие конфигурационные файлы, сконцентрированные на одной задаче. Такие файлы легко переместить или создать ссылку на конфигурационный каталог других служб.

Чтобы применить изменения, внесенные в юнит, выполните следующую команду с привилегиями суперпользователя root:

```
systemctl daemon-reload
systemctl restart name.service
```

Пример: расширение конфигурации httpd.service

Чтобы изменить юнит httpd.service так, чтобы при запуске сервера Apache запускался пользовательский сценарий shell, выполните следующие шаги.

1) Создайте каталог и пользовательский файл конфигурации:

```
# mkdir /etc/systemd/system/httpd.service.d/
# touch /etc/systemd/system/httpd.service.d/custom_script.conf
```

2) Учитывая, что сценарий, который вы хотите автоматически запускать вместе с Apache, расположен по пути /usr/local/bin/custom.sh, введите следующий тест в файл custom_script.conf:

```
[Service]
ExecStartPost=/usr/local/bin/custom.sh
```

3) Чтобы применить изменения, выполните:

```
# systemctl daemon-reload
# systemctl restart httpd.service
```

Примечание. Файлы конфигураций из каталога /etc/systemd/system/ имеют больший приоритет, чем юниты из каталога /usr/lib/systemd/system/. Поэтому, если конфигурационные файлы содержат параметр, который можно указать только один раз, такой как `Description` или `ExecStart`, изначальное значение этого параметра будет предопределено. Обратите внимание, что в выводе команды `systemd-delta`, описанной в п. «Управление переопределенными юнитами» на стр. 184, такие юниты всегда помечены как `[EXTENDED]`, даже если в целом некоторые параметры фактически переопределяются.

Изменение исходной конфигурации юнита

Чтобы внести изменения, которые сохранятся после обновления пакета, содержащего файл юнита, сначала скопируйте этот файл в каталог /etc/systemd/system/. Для этого выполните следующую команду с привилегиями суперпользователя root:

```
# cp /usr/lib/systemd/system/<имя>.service
/etc/systemd/system/<имя>.service
```

Здесь `<имя>` означает имя юнита службы, который нужно изменить. Указанный выше синтаксис применяется ко всем типам юнитов. Откройте скопированный файл в текстовом редакторе и внесите необходимые изменения. Чтобы применить эти изменения, выполните:

```
# systemctl daemon-reload
# systemctl restart name.service
```

Пример: изменение лимита времени ожидания

Во избежание зависания системы по вине неправильно работающей службы для каждой из них можно указать лимит времени ожидания. По умолчанию он равен 90 секундам для обычных служб и 300 секундам — для служб, совместимых с SysV.

Чтобы, например, увеличить значение истечения времени ожидания для службы httpd, выполните следующие действия:

1) Скопируйте файл юнита httpd в каталог /etc/systemd/system/:

```
cp /usr/lib/systemd/system/httpd.service
/etc/systemd/system/httpd.service
```

2) Откройте файл /etc/systemd/system/httpd.service и укажите значение `TimeoutStartUSec` в разделе `[Service]`:

```
...
[Service]
...
PrivateTmp=true
TimeoutStartSec=10

[Install]
WantedBy=multi-user.target
```

```
...
```

3) Перезапустите системный демон:

```
systemctl daemon-reload
```

4) Необязательно: проверьте новое значение истечения времени ожидания:

```
systemctl show httpd -p TimeoutStartUSec
```

Примечание. Чтобы изменить значение истечения времени ожидания глобально, введите DefaultTimeoutStartSec в файле /etc/systemd/system.conf. Смотрите раздел «Введение в systemd»

Управление переопределенными юнитами

Чтобы получить обзорную информацию по переопределенным или измененным файлам юнитов, выполните следующую команду:

```
$ systemd-delta
```

Вывод этой команды может выглядеть таким образом:

```
[EQUIVALENT] /etc/systemd/system/default.target →
/usr/lib/systemd/system/default.target
[OVERRIDDEN] /etc/systemd/system/autofs.service →
/usr/lib/systemd/system/autofs.service

--- /usr/lib/systemd/system/autofs.service      2014-10-16
21:30:39.000000000 -0400
+++ /etc/systemd/system/autofs.service  2014-11-21
10:00:58.513568275 -0500
@@ -8,7 +8,8 @@
 EnvironmentFile=-/etc/sysconfig/autofs
 ExecStart=/usr/sbin/automount $OPTIONS --pid-file
/run/autofs.pid
 ExecReload=/usr/bin/kill -HUP $MAINPID
-TimeoutSec=180
+TimeoutSec=240
+Restart=Always

 [Install]
 WantedBy=multi-user.target

[MASKED]     /etc/systemd/system/cups.service →
/usr/lib/systemd/system/cups.service
[EXTENDED]   /usr/lib/systemd/system/sssd.service →
/etc/systemd/system/sssd.service.d/journal.conf

4 overridden configuration files found.
```

В таблице 13 приведен список типов переопределения, которые могут встретиться в выводе `systemd-delta`. Обратите внимание, что если файл перезаписан, то `systemd-delta` по умолчанию показывает сводку изменений, похожую на вывод команды `diff`.

Таблица 13. Типы различий команды systemd-delta

Тип	Описание
[MASKED]	Замаскированные файлы юнитов, описание см. в п. 14.4.9. «Отключение службы»
[EQUIVALENT]	Копии, в которые не было внесено изменений, но приоритет которых выше приоритета исходных файлов, чаще всего это символьные ссылки
[REDIRECTED]	Файлы, которые перенаправляется на другие файлы
[OVERRIDEN]	Переопределенные и измененные файлы
[EXTENDED]	Файлы, расширенные файлами .conf в каталоге /etc/systemd/system/unit.d/
[UNCHANGED]	Неизмененные файлы показываются, только если был указан параметр --type=unchanged

Рекомендуется выполнять systemd-delta после обновления системы, чтобы проверить, нет ли обновлений для юнитов по умолчанию, которые на данный момент являются переопределенными в пользовательской конфигурации. Также можно ограничить вывод только определенным типом различия. Чтобы, например, просмотреть только переопределенные юниты, выполните:

```
$ systemd-delta --type=overridden
```

Работа с юнитами, имеющими более одного экземпляра

Из одного файла конфигурационного шаблона можно создать несколько юнитов. Для отметки шаблона и для связи юнитов с ним используется символ @. Экземпляр юнита можно запустить из другого файла юнита с помощью параметров Requires или Wants или с помощью команды systemctl. Экземпляры юнитов именуются следующим образом:

```
<имя_шаблона>@<имя_экземпляра>.service
```

Здесь <имя_шаблона> означает имя конфигурационного файла шаблона. Замените <имя_экземпляра> именем экземпляра юнита. Несколько экземпляров могут указывать на один и тот же файл шаблона, с параметрами, общими для всех экземпляров юнита. Имя юнита шаблона имеет следующий вид:

```
<имя_юнита>@.service
```

Следующий параметр Wants в файле юнита (Wants=getty@ttyA.service, getty@ttyB.service) сначала заставляет systemd искать юниты указанной службы. Если таковые не были найдены, часть, расположенная между @ и суффиксом, обозначающим тип, игнорируется, и systemd начинает поиск файла getty@.service, читает параметры из него и запускает службу.

В любом конфигурационном файле юнита можно использовать символы подстановки, которые называются спецификаторами юнита. Спецификаторы юнита заменяют некоторые параметры юнита и интерпретируются во время выполнения. В таблице 14 приведен список спецификаторов юнитов, особенно удобных при работе с юнитами шаблонов.

Таблица 14. Важные спецификаторы юнитов

Спецификатор юнита	Значение	Описание
`%n`	Полное имя юнита	Обозначает полное имя юнита, включая суффикс типа. `%N` имеет такое же значение, но помимо прочего заменяет запрещенные символы кодами ASCII
`%p`	Имя префикса	Обозначает имя юнита без суффикса типа. Для юнитов, имеющих более одного экземпляра, `%p` означает часть имени юнита до символа `@`
`%i`	Имя экземпляра	Часть имени юнита, имеющего более одного экземпляра, между символом `@` и суффиксом типа. `%I` имеет такое же значение, но помимо прочего заменяет запрещенные символы кодами ASCII
`%H`	Имя хоста	Обозначает имя хоста работающей системы на тот момент времени, когда загрузилась конфигурация юнита
`%t`	Каталог времени выполнения	Представляет каталог времени выполнения. Это либо /run для суперпользователя root, либо значение переменной XDG_RUNTIME_DIR для непривилегированных пользователей

Полный список спецификаторов юнитов см. на странице руководства systemd.unit(5).

Шаблон getty@.service, например, содержит следующие директивы:

```
[Unit]
Description=Getty on %I
...
[Service]
ExecStart=-/sbin/agetty --noclear %I $TERM
...
```

При создании экземпляров `getty@ttyA.service` и `getty@ttyB.service` из вышеуказанного шаблона параметр `Description=` разрешается как `Getty` в `ttyA` и `Getty` в `ttyB`.

Доступная локально документация по systemd

- systemctl(1) — страница руководства для консольной утилиты systemctl, предоставляющая полный список поддерживаемых команд и параметров;
- systemd(1) — страница руководства системы systemd и service manager предоставляет больше сведений об устройстве и принципах работы, и документирует доступные консольные параметры и переменные окружения, поддерживаемые конфигурационные файлы и каталоги, распознаваемых сигналах и доступных параметрах

ядра;

- systemd-delta(1) — страница руководства для утилиты systemd-delta, где можно найти расширенные и переопределенные файлы конфигураций;
- systemd.unit(5) — подробная информация о файлах юнитов systemd, а также все доступные параметры конфигураций;
- systemd.service(5) — формат файлов юнитов служб;
- systemd.target(5) — формат файлов юнитов целей;
- systemd.kill(5) — конфигурация действий по завершению процессов.

Как написать файл юнита службы с принудительным запуском конкретных служб

Вопрос:

Параметры `After=/Before=` не гарантируют того, что указанные юниты были фактически запущены и работают. Как написать файл юнита службы, который принуждал бы запуск конкретных служб?

Решение:

Немного теории

В ОС РОСА «КОБАЛЬТ» в файле юнита службы зависимости указываются с помощью параметров `After=/Before=`. Они гарантируют, что в случае их использования служба b.service будет запущена, например, после службы a.service. Но эти параметры не гарантируют того, что служба b.service будет запущена только после успешного запуска службы a.service, т. к. параметры `After=/Before=` только указывают порядок зависимостей, но не учитывают того, запущена ли фактически зависимость или нет.

Если службе необходимо выполнение другой службы-зависимости, необходимо, чтобы параметры, указанные ниже, отвечали этому требованию.

Подробные сведения

В разделе `[Unit]` файла конфигурации юнита могут содержаться следующие параметры, принудительно выполняющие требование успешного запуска и выполнения для перечисленных юнитов (приводимые здесь подробности взяты со страницы руководства systemd.unit(5)):

```
Requires=
Requisite=
PartOf=
```

Подробности параметра `Requires=`

Этот параметр указывает требуемые юниты-зависимости. Если юнит, для которого пишется файл, будет включен, указанные здесь юниты также будут включены. Если один из этих юнитов будет отключен, или если его включение окончится неудачей, юнит, для которого пишется файл, также будет отключен. Этот параметр можно указывать больше одного раза, равно как и для единожды указанного параметра можно указать несколько юнитов, разделенных пробелом; в итоге будут созданы зависимости в виде каждого указанного имени юнита.

Обратите внимание, что требуемые зависимости не влияют на порядок, в котором

запускаются или останавливаются службы. Порядок запуска необходимо настраивать отдельно, с помощью параметров `After=` или `Before=`. Если юниту foo.service требуется юнит bar.service, что настроено при помощи параметра `Requires=`, но порядок запуска не настроен (при помощи параметров `After=` или `Before=`), то в случае, если foo.service включен, оба юнита будут запущены одновременно без какого-либо временного промежутка между ними. Часто для того, чтобы получить систему, более устойчивую к неправильной работе служб, лучше использовать параметр `Wants=` вместо параметра `Requires=`.

Подробности параметра `Requisite=`

Этот параметр аналогичен соответствующему `Requires=`, но если указанные для этого параметра юниты не были запущены, запускаться они не будут, и транзакция немедленно окончится неудачей.

Подробности параметра `PartOf=`

Этот параметр настраивает зависимости аналогично параметру `Requires=`, но ограничен остановкой и запуском юнитов. Когда systemd останавливает или перезапускает перечисленные здесь юниты, действие распространяется и на юнит, для которого пишется файл. Обратите внимание, что это однонаправленная зависимость — изменения юнита, для которого пишется файл, не затрагивают юниты, указанные в параметре.

Как выбрать параметр, подходящий для создаваемого файла юнита?

В данном пункте предлагаются некоторые решения, как выбрать соответствующие ключевые параметры для файла юнита. Далее предполагается, что `a.service` и `b.service` имеют указанные ниже зависимости.

```
a.service (Before=b.service)
b.service (After=a.service)
```

1. Требование: когда включается `b.service`, должен включаться и `a.service` (если он не был включен ранее). Запуск юнита `a.service` должен быть успешным — при неудачном запуске, или если этот юнит не был включен, `b.service` также будет отключен.

Решение: для включения `a.service` необходимо использовать `Requires=`, а также нужно обеспечить, чтобы `b.service` не мог запускаться без запуска `a.service`.

2. Требование: если `a.service` еще не запущен, `b.service` не должен запускаться, и его запуск немедленно должен заканчиваться неудачей. Включение `a.service` не должно срабатывать при включении `b.service`.

Решение: для выполнения условия нужно использовать `Requisite=`.

3. Требование: после включения `a.service` всегда должен включаться `b.service`. Но `a.service` не должен включаться при включении `b.service`, а также `a.service` не должен останавливаться или перезапускаться, если останавливается или перезапускается `b.service`.

Решение: в такой ситуации нужно использовать `PartOf=`.

Для достижения желаемого результата может потребоваться несколько парамет-

ров. Например, для b.service указан параметр Requires=a.service, т. е. b.service не может быть запущен без запуска a.service, и запуск b.service сначала запускает a.service. А также для определения a.service может понадобиться параметр PartOf=b.service, чтобы в случае остановки или перезапуска b.service для a.service выполнялось бы точно такое же действие.

Между юнитами можно настроить взаимозависимости любой сложности, но это требует четкого понимания требуемых взаимодействий. Другие параметры см. на странице руководства systemd.unit(5).

Как решить, какие зависимости должно иметь определение юнита службы systemd

Вопрос

Выяснение того, какие зависимости должно иметь определение юнита службы systemd, является достаточно сложной проблемой, решение которой потребует некоторых усилий. В данном фрагменте делается попытка объяснить некоторые аспекты данной проблемы.

Решение

Немного теории

В отличие от предыдущих версий, где сценарии начала и завершения работы выполнялись последовательно, согласно символическим ссылкам start/stop, в текущей версии ОС РОСА «КОБАЛЬТ» многие службы могут запускаться параллельно, согласно явно настроенным зависимостям.

Здесь возникает проблема того, что при наличии точной информации о зависимостях, определенных с помощью параметров After=, Before= и WantedBy=, службы могут запускаться не так, как ожидается. Информацию о применении других параметров зависимостей, таких как Requisite=, BindsTo=, Conflicts=, PartOf=, Wants=, Requires= и др., см. на странице руководства system.unit или в п. 14.4.24. «Создание и изменение файлов юнитов systemd».

Подробная информация

After=/Before=

Параметр After= обратен параметру Before=. Если юнит указывается в параметре After=, для другого юнита автоматически предполагается то же самое. Таким образом, если файл a.service содержит After=b.service, результат будет аналогичным действию параметра Before=a.service в файле b.service.

Это означает, что a.service не может быть запущен до тех пор, пока b.service не закончит работу. Во время выключения системы эти зависимости выполняются в обратном порядке. Т. е. для того, чтобы b.service мог быть остановлен, необходимо сначала остановить a.service (если он выполняется).

Если в обоих файлах (a.service и b.service) вместо этого присутствует Before=multi-user.target, эти службы должны начать и завершить работу до того, как будет достигнута цель multi-user.target. Тем не менее, в данном случае они могут запускаться одновременно, т. к. больше не имеют никакой явной зависимости друг от дру-

га.

Важно понимать, что параметры `After=/Before=` не различают, была ли служба запущена удачно или неудачно. Информация, передаваемая параметрами `After=/Before=`, касается только порядка, в котором служба останавливается или запускается.

`WantedBy=`

Параметр `WantedBy=` может быть использован более одного раза, или же для одного параметра может указываться список имен юнитов, разделенных пробелами. Во время установки этих юнитов с помощью команды `systemctl enable` в каталоге .wants/ каждого из этих юнитов создается символьная ссылка. Как результат, зависимость типа `Wants=` добавляется от юнита в списке к текущему юниту. Основным результатом является то, что текущий юнит (тот, для которого создается файл) будет запускаться при запуске юнита, указанного в списке.

Если специфические требования к запуску службы на ранних этапах загрузки системы отсутствуют, в раздел `[Install]` файлов определения службы всегда рекомендуется помещать следующую запись:

```
WantedBy=multi-user.target
```

Если, например, файл юнита `a.service` содержит этот параметр, и эта служба включена, для цели multi-user.target будут определены зависимости `Wants=a.service` и `After=a.service`. Это означает, что параметр `WantedBy=multi-user.target` обеспечивает запуск юнита, в файле которого этот параметр содержится (если юнит не отключен и не замаскирован), до того, как будет достигнута цель multi-user.target.

Особые рекомендации

1. Порядок зависимостей должен быть как можно более простым и ясным. При наличии, например, трех служб, которые должны запускаться последовательно, порядок должен быть следующим:

```
a.service (Before=b.service)
b.service (After=a.service and Before=c.service)
c.service (After=b.service)
```

Добавление параметра `Before=/After=` в случае необходимости создает автономные определения юнитов служб, и, просматривая один из них, можно представить весь порядок выполнения. Если, например, рассмотреть `b.service` в примере выше, можно увидеть, что запуск этого юнита не будет выполнен до тех пор, пока не будет запущен `a.service`, а запуск юнита `c.service` будет выполнен только после запуска `b.service`.

Для понимания зависимости рассмотрим несколько примеров. Это файлы юнитов служб располагаются в каталоге /etc/systemd/system.

`* a.service`

```
[Unit]
Description=Test "a" Service
Before=b.service
```

```
[Service]
ExecStart=/bin/true
Type=oneshot
[Install]
WantedBy=multi-user.target
```

*** b.service**

```
[Unit]
Description=Test "b" Service
After=a.service
Before=c.service
[Service]
ExecStart=/bin/true
Type=oneshot
[Install]
WantedBy=multi-user.target
```

*** c.service**

```
[Unit]
Description=Test "c" Service
After=b.service
[Service]
ExecStart=/bin/true
Type=oneshot
[Install]
WantedBy=multi-user.target
```

Очевидно, что эти службы служат только для иллюстрации следующих команд. Теперь мы видим, что нужно запустить до того, как будет запущен `b.service`, и что будет запущено после него:

```
# systemctl list-dependencies --after b.service
b.service
├─a.service
├─system.slice
├─systemd-journald.socket
└─basic.target
   ├─rhel-import-state.service
   ├─systemd-ask-password-plymouth.path
   ├─paths.target
   │ ├─brandbot.path
   │ ├─systemd-ask-password-console.path
   │ └─systemd-ask-password-wall.path
   ├─slices.target
```

Параметр `--after` команды `systemctl list-dependencies` показывает, что

именно указано в качестве зависимости для службы, указанной в аргументе команды. Иными словами, какие юниты имеют зависимость `Before=` относительно юнита `b.service`. Что именно будет запущено после `b.service`, в выводе команды не показывается. Мы не ожидаем увидеть в этом выводе юнит `a.service`, т. к. в его определении явно содержится зависимость `Before=` относительно `b.service`. Другие зависимости являются автоматическими.

И наоборот, с помощью параметра `--before` мы можем просмотреть, для каких юнитов указана зависимость `After=` относительно юнита `b.service`.

```
# systemctl list-dependencies --before b.service
b.service
├─c.service
├─multi-user.target
│ ├─systemd-readahead-done.service
│ ├─systemd-readahead-done.timer
│ ├─systemd-update-utmp-runlevel.service
│ └─graphical.target
│
└─systemd-update-utmp-runlevel.service
   └─shutdown.target
   ├─systemd-reboot.service
   └─final.target
   └─systemd-reboot.service
```

Эта команда может помочь в понимании порядка работы служб в systemd. Если у службы нет зависимости от другой службы, то она будет запущена параллельно до того, как будет достигнута присвоенная службе цель, что определено с помощью параметров `Before=`, `After=` или `WantedBy=`. Обратите внимание, что цель `shutdown.target` добавляется автоматически, т. к. указанная служба должна быть остановлена до того, как будет достигнута цель `shutdown.target`.

2. Использование `After=network-online.target`. Этот параметр указывает, чтобы служба запускалась только после того, как будет поднята и доступна сеть. Подробности см. по ссылке https://www.freedesktop.org/wiki/Software/systemd/NetworkTarget (англ.). Обычно это требование не нужно, если только сценарий запуска не требует доступа к активной сети (ожидание network-online.target может замедлить общий процесс загрузки системы).

Обратите внимание, что цель `network-online.target` отсутствует в процессе выключения системы. Чтобы эта цель вызывалась во время выключения системы, до отключения сети, необходимо добавить `network.target` в параметр `After=`. Это условие предполагает, что для выключения службе необходимо активное сетевое соединение.

Известные ошибки, которые могут быть обусловлены зависимостями порядка

Использование `After=<something>.target` (где `<something>` заменяется на действительное имя цели). Некоторые команды могут работать нестабильно до того мо-

мента, когда будет достигнута цель `sysinit.target`. Пример: команда `rpm`, которая может работать нестабильно до того момента, пока не будет достигнута цель `sysinit.target`.

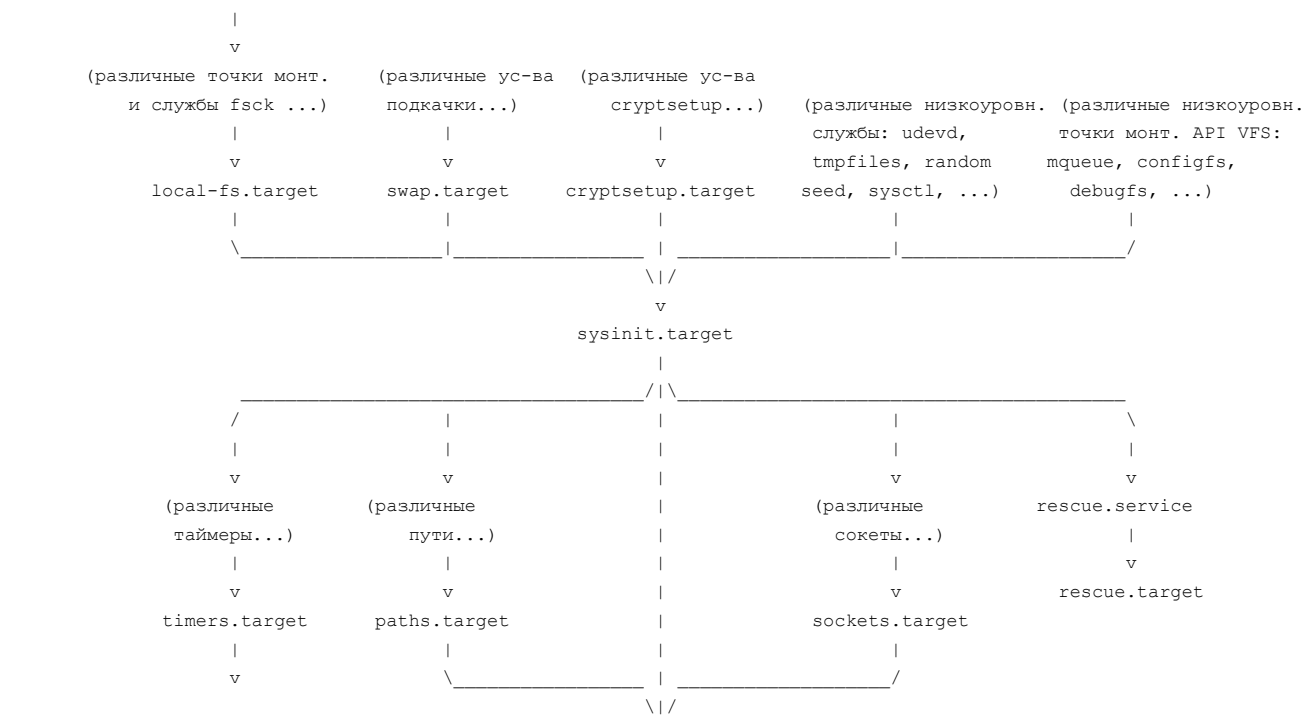
Ниже приведен обзор последовательности загрузки, так, как она описывается на странице руководства bootup(7). Подробности о каждой конкретной цели, указанной в обзоре ниже, можно найти на странице руководства systemd.special(7).

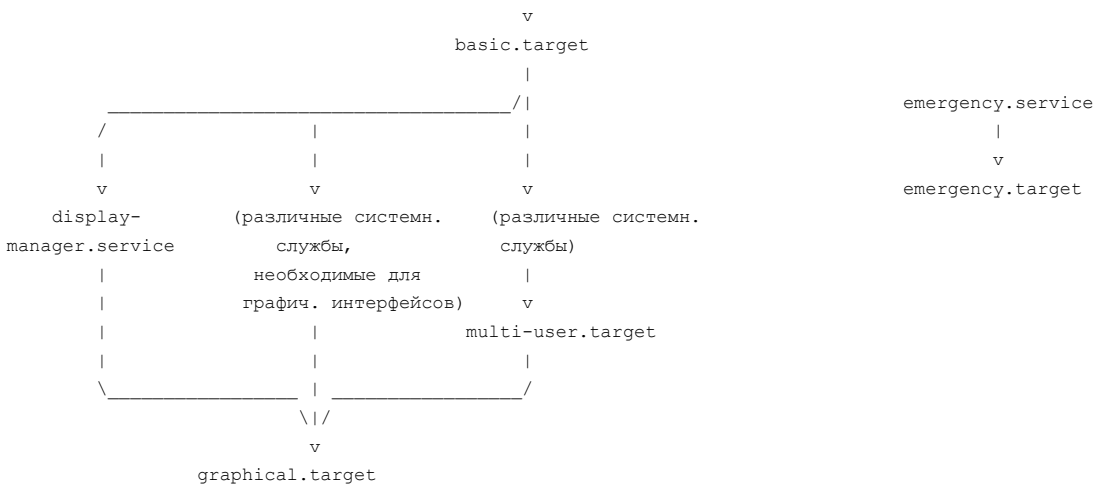
Процесс загрузки system manager

Во время загрузки system manager несет ответственность за инициализацию файловых систем, служб и драйверов, необходимых для работы системы. В системах на базе systemd(1) этот процесс разделен на несколько отдельных этапов, которые представлены в виде юнитов целей. (Подробности о юнитах целей см. на странице руководства systemd.target(5) и в п. 14.4.11. «Работа с целями systemd»). Процесс загрузки является крайне параллелизованным, поэтому порядок, в котором достигаются отдельные цели, не является жестко обусловленным, но все-таки привязан к некоторой минимальной упорядочивающей структуре.

Когда systemd запускает систему, он включает все юниты, являющиеся зависимостями `default.target` (а также рекурсивные зависимости этих зависимостей). Обычно `default.target` — это просто псевдоним для `graphical.target` или `multi-user.target`, в зависимости от того, настроена ли система на работу с графическим интерфейсом или в консольном режиме. Для принудительного минимального упорядочивания участвующих в процессе юнитов существует некоторое число хорошо известных юнитов целей, указанных на странице руководства systemd.special(7).

Ниже приведено графическое представление структуры этих хорошо известных юнитов и их места в логической схеме процесса загрузки. Стрелки показывают, какие юниты участвуют в процессе, и на каком месте они стоят в порядке запуска. Юниты, расположенные ближе к верху схему, запускаются раньше юнитов, расположенных ниже.





Здесь делается акцент на юниты целей, которые обычно используются как цели загрузки. Эти юниты удобно выбирать в качестве целей, которые нужно достичь, передавая их, например, консольному параметру ядра `systemd.unit=` (см. systemd(1)) или создавая символьную ссылку на них с default.target. Юнит timers.target задействуется юнитом basic.target асинхронно, что позволяет юнитам таймеров иметь зависимости от служб, которые будут доступны при загрузке позже.

Настройка лимитов для служб в ОС РОСА «КОБАЛЬТ» и systemd

Вопрос

Как настроить лимиты для служб, запускаемых systemd во время загрузки системы? Лимиты, настроенные в /etc/security/limits.conf или /etc/security/limits.d/*.conf, не работают.

Решение

Определение службы можно расширить, что описывается на странице руководства systemd.unit(5) и в п. 14.4.24. «Создание и изменение файлов юнитов systemd». Используя в качестве примера tftp.service, создайте новый подкаталог tftp.service.d в каталоге /etc/systemd/system, а затем в этом подкаталоге создайте файл conf, расширяющий (или переопределяющий) параметры службы.

В этом примере количество открытых дескрипторов файлов ограничено 500 000.

```
# mkdir -p /etc/systemd/system/tftp.service.d/
# cat >/etc/systemd/system/tftp.service.d/filelimit.conf <<EOF
[Service]
LimitNOFILE=500000
EOF
```

Изменения применяются после перезагрузки конфигурации демона и перезапуска службы.

```
# systemctl daemon-reload
# systemctl restart tftp.service
```

Команды `systemd-delta` и `systemctl status tftp.service` показывают, что определение службы было расширено.

```
# systemd-delta --type=extended
[EXTENDED]
```

```
/usr/lib/systemd/system/tftp.service → /etc/systemd/system
/tftp.service.d/filelimit.conf
1 overridden configuration file found.
# systemctl status tftp.service
● tftp.service - Tftp Server
Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect;
vendor
preset: disabled)
Drop-In: /etc/systemd/system/tftp.service.d
└─filelimit.conf
...
```

Возможные лимиты описаны в следующих разделах страницы руководства systemd.exec(5):

```
LimitCPU=, LimitFSIZE=, LimitDATA=, LimitSTACK=, LimitCORE=,
LimitRSS=,
LimitNOFILE=, LimitAS=, LimitNPROC=, LimitMEMLOCK=, LimitLOCKS=,
LimitSIGPENDING=, LimitMSGQUEUE=, LimitNICE=, LimitRTPRIO=,
LimitRTTIME=

These settings control various resource limits for executed
processes. See setrlimit(2) for details. Use the string infinity
to
configure no limit on a specific resource.
```

Причина

Лимиты, указанные в файлах /etc/security/limits.conf или /etc/security/limits.d/*.conf, настраиваются PAM в начале сеанса регистрации, что указано в следующей записи файла /etc/pam.d/system-auth-ac:

```
session        required        pam_limits.so
```

Поскольку демоны, запускаемые systemd, не используют сеанс регистрации PAM, лимиты можно настраивать только в файле юнита службы.

15. УПРАВЛЕНИЕ ПЕЧАТЬЮ

15.1. Служба CUPS и консольная утилита lpadmin

CUPS (Common UNIX Printing System) — основанная на стандартах система печати с открытым кодом, разрабатываемая корпорацией Apple для операционной системы macOS и других ОС на базе UNIX.

15.1.1. Установка CUPS

```
# yum -y install cups
```

15.1.2. Управление службой CUPS

Запуск:

```
# systemctl start cups.service
```

Автоматический запуск службы при загрузке системы:

```
# systemctl enable cups.service
```

Перезапуск службы:

```
# systemctl restart cups.service
```

Просмотр статуса службы:

```
$ systemctl status cups.service
```

Немедленная остановка службы:

```
# systemctl stop cups.service
```

Базовый источник документации в локальной установке службы CUPS расположен по адресу http://localhost:631/help.

15.1.3. Консольная утилита lpadmin

Утилита lpadmin служит для настройки параметров службы печати lp в системах Linux.

lpadmin настраивает очереди принтеров и классов, предоставленных службой печати CUPS. Также lpadmin можно использовать для указания принтера или класса по умолчанию на сервере.

Параметры команды lpadmin:

- -E, указанный перед параметрами -d, -p или -x, служит для принудительного использования шифрования при соединении с сервером;
- -d указывает цель (принтер или класс) по умолчанию. Дальнейшие задачи печати, переданные с использованием команд lp или lpr, будут использовать эту цель до тех пор, пока пользователь не укажет другой принтер с помощью команды lpoptions;
- -p настраивает именованный принтер или класс. Дополнительные параметры описываются ниже;
- -x удаляет цель (принтер или класс) по умолчанию. Любые задачи, стоящие в очереди для этой цели, будут удалены, а текущее задание печати будет аварийно от-

менено.

15.1.4. Синтаксис команды lpadmin

```
lpadmin [ -E ] [-U <имя_пользователя>] [ -h <сервер>[:<порт>] ]
-d <цель>
lpadmin [ -E ] [-U <имя_пользователя>] [ -h <сервер>[:<порт>] ]
-p <цель>
        [ -R <имя_по_умолчанию> ] <параметр(ы)>
lpadmin [ -E ] [-U <имя_пользователя>] [ -h <сервер>[:<порт>] ]
-x <цель>
```

15.1.4.1. Параметры

Для настройки очереди принтера используются следующие параметры:

- `-c <класс>` добавляет именованный принтер к классу. Если класс не существует, он создается автоматически;
- `-i <интерфейс>` настраивает сценарий интерфейса для принтера в стиле System V. Этот параметр нельзя уточнить с помощью параметра -P (файл PPD), и он предназначен для поддержки драйверов старых принтеров;
- `-m <модель>` настраивает стандартный сценарий с интерфейсом System V, файл PPD из каталога моделей или с использованием одного из интерфейсов драйвера. Для получения списка поддерживаемых моделей применяйте команду lpinfo с ключом -m;
- `-o cupsIPPSupplies=true`, `-o cupsIPPSupplies=false` указывает, нужно ли сообщать значения уровня ресурсов IPP (Internet Printing Protocol, «протокол печати через интернет»);
- `-o job-k-limit=value` устанавливает лимит квоты на каждого пользователя (в КБ). Значение — целое число килобайтов, один килобайт = 1024 байт;
- `-o job-page-limit=value` устанавливает лимит страниц для квот на каждого пользователя. Значение — целое число страниц, которое можно напечатать; при двусторонней печати одна сторона = одна страница;
- `-o job-quota-period=value` устанавливает учетный период для квот на каждого пользователя. Значение — целое число секунд; один день равен 86 400 с;
- `-o job-sheets-default=<титул>[, <титул>]` настраивает титульную страницу (или страницы) по умолчанию для задач печати;
- `-o имя=значение` указывает параметр PPD для принтера. Список параметров PPD можно получить с помощью команды `lpoptions -l`;
- `-o <имя_по_умолчанию>=<значение>` указывает серверный параметр по умолчанию для цели. Любой параметр печати можно сделать параметром по умолчанию; чтобы, например, установить значение `cpi` по умолчанию равным 17, используйте `-o cpi-default=17`;
- `-o port-monitor=<имя>` указывает, какую программу бинарной связи нужно ис-

пользовать во время печати — нет, bcp или tbcp. По умолчанию — нет. Указанный монитор порта должен присутствовать в файле PPD;

- -o printer-error-policy=<имя> указывает политику ошибок, используемую, если фоновая программа принтера не может послать задачу на принтер. Имя должно быть одним из следующих: abort-job, retry-job, retry-current-job или stop-printer. Политика ошибок по умолчанию: stop-printer для принтеров, retry-current-job — для классов;
- -o printer-is-shared=[true|false] делает цель общей/опубликованной (shared/published) или недоступной для общего пользования/неопубликованной (unshared/unpublished). Общие/опубликованные цели публично объявляются сервером в LAN на основе параметра Browse в cupsd.conf, а недоступные для общего пользования/неопубликованные цели не объявляются. Значение по умолчанию — true;
- -o printer-op-policy=<имя> указывает политику работы IPP (протокол печати через интернет), связанную с целью. Имя должно быть указано в файле cupsd.conf в разделе Policy. Политика работы по умолчанию — default;
- -R <имя_по_умолчанию> удаляет параметр named для принтера
- -r <класс> удаляет именованный принтер из класса. Если в итоге класс становится пустым, он тоже удаляется.
- -u [allow: пользователь, пользователь, @группа; deny: пользователь, пользователь, @группа; allow: all, deny: none] настраивает контроль доступа пользователей для цели. Имена, начинающиеся с символа @, интерпретируются как группы UNIX. Последние два параметра отключают контроль за доступом на пользовательском уровне;
- -v "device-uri" устанавливает атрибут device-uri для очереди принтера. Для получения списка поддерживаемых адресов URI и схем для устройства используйте команду lpinfo -v;
- -D "info" предоставляет текстовое описание цели;
- -E активирует цель и принимает задачи; аналог выполнения программ cupsaccept и cupsenable для цели;
- -L "<местонахождение>" предоставляет текстовое местонахождение цели;
- -P <файл_ppd> указывает на файл PPD, который нужно использовать для принтера. При наличии этого параметра он имеет больший приоритет, чем параметр -i (сценарий интерфейса).

15.1.4.2. Примеры использования

Примечание. Параметры, используемые в командной строке, нельзя группировать.

Получение списка устройств

```
# lpinfo -v
$ /usr/lib/cups/backend/snmp <адрес_ip> # для нахождения URI используйте SNMP
```

Получение списка моделей

```
$ lpinfo -m
```

Добавление новой очереди

```
# lpadmin -p имя_очереди -E -v uri -m модель
```

Имя очереди определяет пользователь. Пример:

```
# lpadmin -p HP_DESKJET_940C -E -v "usb://HP/DESKJET%20940C?
serial=CN16E6C364BH" -m drv:///HP/hp-deskjet_940c.ppd.gz
# lpadmin -p AirPrint -E -v "ipp://10.0.1.25/ipp/print" -m
everywhere # очередь без драйвера (Apple AirPrint или IPP
Everywhere)
# lpadmin -p SHARED_PRINTER -m raw # простая очередь; без PPD или
фильтра
```

Указание принтера по умолчанию (цели)

```
$ lpoptions -d имя_ очереди
```

Смена параметров

```
$ lpoptions -p имя_ очереди -l # список параметров
$ lpoptions -p имя_ очереди -o option=value # назначение парамет-
ра
```

Пример:

```
$ lpoptions -p HP_DESKJET_940C -o PageSize=A4
```

Проверка статуса

```
$ lpstat -s
$ lpstat -p <имя_очереди>
```

Отключение принтера

```
# cupsdisable <имя_очереди>
```

Включение принтера

```
# cupsenable <имя_очереди>
```

Настройка принтера на принятие задач

```
# cupsaccept <имя_очереди>
```

Удаление принтера

1) Настройте принтер на сброс всех входящих запросов:

```
# cupsreject <имя_очереди>
```

2) Отключите принтер:

```
# cupsdisable <имя_очереди>
```

3) Удалите принтер:

```
# lpadmin -x <имя_очереди>
```

Печать файла

```
$ lpr файл
$ lpr -# 17 файл # напечатать файл 17 раз
$ echo "Hello, world!" | lpr -p # напечатать вывод команды. Пара-
метр -p добавляет заголовок
```

Проверка очереди

```
$ lpq
$ lpq -a # во всех очередях
```

Очистка очереди

```
# lprm   # удалить только последний элемент очереди
# lprm - # удалить все элементы очереди
```

Добавление принтера

Чтобы добавить принтер с именем Laserjet, расположенный в сети по адресу 10.1.1.1., с использованием файла драйвера CUPS laserjet.ppd, выполните:

```
lpadmin -p LaserJet -E -v socket://10.1.1.1 -m laserjet.ppd
```

Дополнительную информацию об использовании утилит командной строки CUPS можно найти в локальной документации по адресу http://localhost:631/help/options.html.

Примеры установки параметров печати в командной строке с помощью lpadmin

Данный параграф содержит ответы на следующие вопросы:

- Как настроить принтер, драйверы которого есть в составе пакетов ОС РОСА «КОБАЛЬТ», с использованием командной строки?
- Как с помощью консоли добавить очередь печати, которая бы указывала на последовательное устройство?
- Как добавить принтер(ы) в сервер печати CUPS без использования графических утилит?
- У нас в организации настроены сотни принтеров, но на данный момент мы имеем неупорядоченную смесь из сокетов/LPD, имен/адресов IP и различных драйверов. Как нам стандартизировать и упорядочить все имеющиеся принтеры и их параметры с использованием консольных команд?

Решение

Если для устанавливаемого принтера уже имеется файл PPD (PostScript Printer Definition), переходите к шагу 4.

1) Установите самые свежие пакеты foomatic и hplip:

   ```
   yum -y install foomatic hplip
   ```

2) Получите список поддерживаемых принтеров с помощью команды lpinfo:

   ```
   lpinfo -m
   ```

   Пример: требуется получить список файлов PPD, доступных для принтера Ricoh Aficio MP 2000.

   ```
   # lpinfo -m | grep -i 'aficio.*2000'
   foomatic-db-ppds/Ricoh/PS/Ricoh-Aficio_CL2000_PS.ppd.gz Ricoh
   Aficio
   CL2000 PS
   foomatic-db-ppds/Ricoh/PS/Ricoh-Aficio_MP_2000_PS.ppd.gz Ricoh
   Aficio MP
   2000 PS
   foomatic-db-ppds/Ricoh/PXL/Ricoh-Aficio_MP_2000_PXL.ppd.gz Ricoh
   Aficio
   MP 2000 PXL
   ```

3) Для получения списка доступных устройств печати воспользуйтесь командой lpinfo. Например:

```
# lpinfo -v
network socket
network https
network ipps
network ipp
network http
network lpd
direct hp
serial serial:/dev/ttyS0?baud=115200
network beh
direct hpfax
network smb
```

4) Соберите все в одну очередь печати с помощью команды lpadmin:

```
lpadmin -p <имя_очереди_печати> -m <модель_из_lpinfo> -v <device-uri> -E
```

или

```
lpadmin -p <имя_очереди_печати>-P </путь/до/файла/ppd/> -v <uri_устройства> -E
```

Здесь:

- `-p <имя_очереди_печати>` — имя очереди печати, которую нужно настроить;
- `-m <модель_из_lpinfo>` — информация о модели принтера, возвращенная командой `lpinfo -m`;
- `-P </путь/до/файла/ppd/>` — имеющийся файл PPD;
- `-v <uri_устройства>` — действительный адрес URI устройства, созданный на основе информации, возращенной командой `lpinfo -v`;
- `-E` — команда включения принтера.

Пример: требуется установить ранее упомянутый принтер Ricoh Aficio MP 2000 (предположим, что его адрес IP равен 10.1.2.3, а очередь печати называется «rpm2000»):

```
lpadmin -m -P rmp2000 -m foomatic-db-ppds/Ricoh/PS/Ricoh-Aficio_MP_2000_PS.ppd.gz -v socket://10.1.2.3/ -E
```

Дополнительную информацию о том, где искать и как использовать файлы PPD, см. в подразделе 15.2. «Установка файлов PPD, отсутствующих в репозиториях ОС РОСА «КОБАЛЬТ».

Примеры

Сеть — JetDirect/AppSocket

```
# lpadmin -p 5th-floor-mfp -v socket://10.1.2.3:9100 -m foomatic-db-ppds/Ricoh/PS/Ricoh-Aficio_CL2000_PS.ppd.gz -E
```

Сеть — LPD

```
# lpinfo -m | grep Canon | grep imageRunner | grep 'C6800'
```

```
foomatic:Canon-imageRunner_C6800-hpijs-pcl5c.ppd Canon
imageRunner C6800
Foomatic/hpijs-pcl5c
foomatic:Canon-imageRunner_C6800-Postscript.ppd Canon imageRunner
C6800
Foomatic/Postscript

# lpadmin -p canon-west -v lpd://10.1.2.3/PASSTHRU -m
foomatic:Canon-
imageRunner_C6800-hpijs-pcl5c.ppd -E
```

Подробности о том, как выполнять печать с помощью принтеров LPD, см. ниже в разделе «Настройка серверов печати Unix в ОС РОСА «КОБАЛЬТ».

USB

```
# lpinfo -m | grep Epson | grep Photo
drv:///sample.drv/stphoto2.ppd Epson New Stylus Photo Series
foomatic:Epson-Stylus_Photo_750-stcolor.ppd Epson Stylus Photo
750
Foomatic/stcolor
drv:///sample.drv/stphoto.ppd Epson Stylus Photo Series

# lpadmin -p local-epson-photo -E -v usb:/dev/usb/lp0 -m
drv:///sample.drv/stphoto2.ppd
```

Последовательный порт

```
# lpinfo -v | grep serial
serial serial:/dev/ttyS0?baud=115200

# lpinfo -m | grep Epson | grep Dot
foomatic:Epson-Dot_Matrix-eps9high.ppd Epson Dot Matrix
Foomatic/eps9high
foomatic:Epson-Dot_Matrix-eps9mid.ppd Epson Dot Matrix
Foomatic/eps9mid
foomatic:Epson-Dot_Matrix-epson.ppd Epson Dot Matrix
Foomatic/epson
foomatic:Epson-Dot_Matrix-epsonc.ppd Epson Dot Matrix
Foomatic/epsonc

# lpadmin -p local-dot-matrix -E -v serial:/dev/ttyS0?baud=115200
-m
foomatic:Epson-Dot_Matrix-epson.ppd
```

Параллельный порт

```
# lpinfo -m | grep Epson | grep Photo
   drv:///sample.drv/stphoto2.ppd Epson New Stylus Photo Series
   foomatic:Epson-Stylus_Photo_750-stcolor.ppd Epson Stylus Photo
750
```

```
Foomatic/stcolor
   drv:///sample.drv/stphoto.ppd Epson Stylus Photo Series
# lpadmin -p local-epson-photo -E -v usb:/dev/usb/lp0 -m
foomatic:Epson-Stylus_Photo_750-stcolor.ppd
```

Samba/принтеры Windows

```
# lpinfo -m | grep HP | grep LaserJet | grep 8150
   foomatic:HP-LaserJet_8150-lj4dith.ppd HP LaserJet 8150
Foomatic/lj4dith
   foomatic:HP-LaserJet_8150-lj5gray.ppd HP LaserJet 8150
Foomatic/lj5gray
   foomatic:HP-LaserJet_8150-ljet4.ppd HP LaserJet 8150
Foomatic/ljet4
   foomatic:HP-LaserJet_8150-Postscript.ppd HP LaserJet 8150
Foomatic/Postscript
   foomatic:HP-LaserJet_8150-pxlmono.ppd HP LaserJet 8150
Foomatic/pxlmono

# lpadmin -p winprinter -E -v smb://username:password@10.1.2.3/HP
-m
foomatic:HP-LaserJet_8150-Postscript.ppd
```

15.2. Установка файлов PPD, отсутствующих в репозиториях ОС РОСА «КОБАЛЬТ»

Данный подраздел содержит ответы на следующие вопросы:

- Как установить сторонние файлы PPD (PostScript Printer Definition, «описание принтера PostScript»), если при добавлении нового принтера через веб-интерфейс или графическую утилиту его модель отсутствует?
- Как установить новый принтер, если для него отсутствует файл PPD в списке доступных драйверов?

15.2.1. Что такое PPD

Описание принтера PostScript (PPD) — это файлы конфигурации, которые сообщают системе печати CUPS сведения о том, как преобразовывать документы в формат, воспринимаемый принтером. Файлы PPD также передают системе печати CUPS доступные для печати параметры (приемный лоток, размер бумаги, параметры сшивания и т. д.). Файлы PPD в Linux иногда называются драйверами принтера.

15.2.2. Источники файлов PPD

Файлы печати PPD, не включенные в комплект поставки ОС РОСА «КОБАЛЬТ», можно скачать из разных источников, в том числе:

База данных принтеров проекта Open Printing

1) Зайдите в базу данных проекта Open Printing по адресу http://www.openprinting.org/printers, выберите производителя имеющегося принтера

и нажмите на кнопку [Показать все].

2) Выберите принтер из списка поддерживаемых моделей. Если точная модель имеющегося принтера не указана, можно выбрать наиболее близкую по характеристикам модель. Если, например, в наличии имеется модель C5502, выбор C5000 может дать рабочую информацию (хотя это не всегда так).
3) Выбор модели открывает страницу, на которой указан уровень поддержки в Linux, доступной для этого принтера. При наличии ссылки на файл PPD скачайте его и сохраните во временном каталоге.

Официальный сайт производителя

Многие производители размещают файлы PPD в разделе загрузок на своем официальном сайте. Файлы PPD часто включаются в архивы (файлы zip или «тарболы») вместе с другим ПО для печати. Если для файла PPD не требуется другого программного обеспечения (что обычно бывает, если искомый принтер поддерживает печать PostScript), можно скачать весь архив и извлечь из него только файл PPD.

15.2.3. Установка файла PPD

1) Скопируйте файл PPD, полученный одним из описанных выше способов, в каталог /usr/share/cups/model.
2) Перезапустите службу CUPS:

```
systemctl restart cups.service
```

После перезапуска службы модель нового принтера должна появиться в списке, доступном в веб-интерфейсе CUPS по адресу http://127.0.0.1:631.

Добавить принтер с новым файлом PPD также можно с помощью следующей команды, запущенной с привилегиями суперпользователя root:

```
# lpadmin -p <имя_очереди> -E -v <протокол>://<IP-адрес_принтера>
-P /usr/share/cups/model/<имя_файла_ppd>.ppd
```

Здесь:

- <имя_очереди> — имя очереди печати, с которой будет связан принтер;
- <протокол> — тип протокола, используемого для связи с принтером (обычно это сокет для принтеров, подключенных по сети, lpd — для принтеров, подключенных через LPD или smb — для принтеров, подключенных к системам Windows);
- <IP-адрес_принтера> — адрес IP или имя хоста принтера, подключенного по сети;
- <имя_файла_ppd> — имя файла PPD, который был сохранен в каталог /usr/share/cups/model.

Пример: как настроить главный сервер печати с использованием CUPS в ОС РОСА «КОБАЛЬТ»?

Необходимо настроить главный сервер печати CUPS для создания общих принтеров для клиентов в нашей сети. Пользователь непосредственного сервера печати может выполнять печать на любом настроенном принтере, но при попытках печати с клиентов выводится ошибка «lp: Connection refused».

Решение

Создание главного сервера печати CUPS

1) Остановите выполнение службы CUPS и создайте резервную копию текущих параметров CUPS:

```
# service cups stop
# cp -a /etc/cups /etc/cups.saved
```

2) Внесите изменения в файл /etc/cups/cupsd.conf, разрешающие другим серверам подключаться к принтерам на главном сервере. Сначала поменяйте Listen localhost:631 на Listen *:631 или на Listen 0.0.0.0:631. Это действие настроит CUPS на прослушивание всех сетевых интерфейсов. Если нужно ограничить прослушивание CUPS каким-то конкретным интерфейсом, введите адрес этого интерфейса вместо символа * или адреса 0.0.0.0. Например:

```
Listen 192.168.102.32:631
```

3) В конец записи <Location /> добавьте Allow @LOCAL. Запись должна выглядеть следующим образом:

```
<Location />
# Allow shared printing
Order allow,deny
Allow @LOCAL
</Location>
```

Этот параметр даст возможность клиентам в сети «local» (подсеть, в которой расположен главный сервер печати) получить доступ к службе CUPS на главном сервере печати. Если доступ к главному серверу печати необходимо разрешить всем клиентам, вместо Allow @Local используйте Allow all.

4) Если CUPS должна посылать широковещательные пакеты с информацией об общих спринтерах, убедитесь, что в файле /etc/cups/cupsd.conf присутствуют следующие записи (эти записи имеются в версии файла по умолчанию):

```
Browsing On
BrowseLocalProtocols cups dnssd
```

Если сервер CUPS не должен рассылать широковещательные пакеты, а вместо этого клиенты CUPS должны опрашивать серверы CUPS на наличие общих принтеров, внесите в файл /etc/cups/cupsd.conf следующую запись:

```
Browsing Off
```

5) Убедитесь, что каждая из очередей печати является общей. Действие по умолчанию — очередь делается общей при ее создании, поэтому проблем возникнуть не должно. Статус общей доступности очереди печати можно проверить, выполнив команду lpoptions и просмотрев вывод на наличие параметра printer-is-shared:

```
# lpoptions -p textonly
copies=1 device-uri=socket://10.3.4.5/ finishings=3 job-hold-
until=no-hold job-priority=50 job-sheets=none,none
marker-change-time=0 number-up=1 printer-commands=none printer-
info=textonly printer-is-accepting-jobs=true
printer-is-shared=true printer-location printer-make-and-
model='Generic text-only printer' printer-state=3
```

```
printer-state-change-time=1478741330
printer-state-reasons=none printer-type=4100
printer-uri-supported=ipp://localhost:631/printers/textonly
```

В этом выводе обратите внимание на запись printer-is-shared=true в третьей строчке.

6) Убедитесь, что служба avahi-daemon установлена и выполняется. Это можно сделать с помощью следующих команд:

```
# yum -y install avahi
# systemctl enable avahi-daemon
# systemctl start avahi-daemon
```

7) Запустите службу CUPS:

```
# systemctl restart cups.service
```

8) Проверьте вывод команды lpstat -t и убедитесь, что на главном сервере печати эти принтеры определены и активированы:

```
# lpstat -t
scheduler is running
no system default destination
device for pcl: socket://10.1.2.3/
device for postscript: socket://10.2.3.4/
device for textonly: socket://10.3.4.5/
pcl accepting requests since Wed 09 Nov 2016 05:28:19 PM PST
postscript accepting requests since Wed 09 Nov 2016 05:28:37 PM
PST
textonly accepting requests since Wed 09 Nov 2016 05:28:50 PM PST
printer pcl is idle. enabled since Wed 09 Nov 2016 05:28:19 PM
PST
printer postscript is idle. enabled since Wed 09 Nov 2016
05:28:37 PM
PST
printer textonly is idle. enabled since Wed 09 Nov 2016 05:28:50
PM PST
```

9) Если на главном сервере печати работает межсетевой экран, администратор должен разрешить внешний доступ к порту 631/ipp, а также к порту 5353/mdns в новой версии ОС РОСА «КОБАЛЬТ» для протоколов UDP и TCP.

Настройка клиентов CUPS

Следующие шаги включают внесение изменений в конфигурационный файл CUPS /etc/cups/cups-browsed.conf.

Вносимые изменения зависят от условий конкретного окружения. Если клиенты должны активно опрашивать сервер CUPS на наличие информации от общих принтеров, или же если клиенты и главный сервер печати находятся в разных подсетях, в соответствующий конфигурационный файл CUPS необходимо добавить следующую запись, заменив текущий адрес IP (или имя хоста) главного сервера печати на адрес 10.12.13.14:

```
BrowsePoll 10.12.13.14
```

При наличии в окружении нескольких главных серверов печати (как, например, в случае настройки CUPS с высокой доступностью) для каждого из них используется отдельная запись BrowsePoll. Поскольку параметр BrowsePoll активно опрашивает главный сервер на наличие информации (с использованием запроса IPP CUPS-Get-Printers), этот способ действует для всех подсетей. Тем не менее, поскольку для опроса сервера требуется подключение TCP, в итоге потребляемый объем сетевых ресурсов будет чуть выше, чем при использовании способа, описываемого далее.

Если клиенты должны пассивно ожидать широковещательной информации от главного сервера печати, в конфигурационный файл CUPS нужно добавить следующую запись:

```
BrowseRemoteProtocols dnssd cups
```

Эти параметры дадут возможность клиентам CUPS собирать информацию об общих принтерах с главного сервера печати в локальной подсети. Для обнаружения общих принтеров здесь используется mDNS/DNS-SD, поэтому сетевые ресурсы будут использоваться чуть менее интенсивно, чем в способе с активным опросом, описанным выше. Тем не менее, этот способ не сработает, если главные сервер печати (или серверы) находится в разных подсетях с клиентами (если только между подсетями не настроен мост mDNS/DNS-SD).

1) Запустите CUPS:

```
# systemctl start cups.service
```

2) Установите, активируйте и запустите службу avahi-daemon:

```
# yum -y install avahi
# systemctl enable avahi-daemon
# systemctl start avahi-daemon
```

3) Активируйте и запустите службу cups-browsed:

```
# systemctl enable cups-browsed
# systemctl start cups-browsed
```

4) Убедитесь, что удаленные принтеры теперь доступны на локальном клиенте:

```
$ lpstat -t
scheduler is running
no system default destination
device for pcl: ipps://master-server.local:631/printers/pcl
device for postscript: ipps://master-print-
server.local:631/printers
/postscript
device for textonly: ipps://master-print-
server.local:631/printers
/textonly
pcl accepting requests since Wed 09 Nov 2016 05:18:09 PM PST
postscript accepting requests since Wed 09 Nov 2016 05:18:09 PM
PST
textonly accepting requests since Wed 09 Nov 2016 05:18:09 PM PST
printer pcl is idle. enabled since Wed 09 Nov 2016 05:18:09 PM
```

```
PST
printer postscript is idle. enabled since Wed 09 Nov 2016
05:18:09 PM
PST
printer textonly is idle. enabled since Wed 09 Nov 2016 05:18:09
PM PST
```

5) Попробуйте выполнить тестовую печать:

```
$ lp -d textonly /etc/fstab
request id is textonly-1 (1 file(s))
```

При правильно настроенных параметрах тестовая печать должна выполниться успешно.

16. НАСТРОЙКА ТИПОВЫХ СЕТЕВЫХ СЛУЖБ

16.1. Настойка сервера NTP

16.1.1. Уровни NTP

Серверы NTP классифицируются согласно дистанции их синхронизации с атомными часами, являющимися источником сигналов времени. Серверы рассматриваются как упорядоченные по уровням (или стратам, Stratum) от самого первого (1) наверху до пятнадцатого (15) в самом низу. Атомные часы считаются нулевым уровнем (0), поскольку являются непосредственным источником времени, но пакеты нулевого уровня никогда не посылаются в интернет. Все атомные часы нулевого уровня привязаны к серверу, который считается уровнем номер 1. Серверы первого уровня посылают в интернет пакеты с пометкой Stratum 1 (первый уровень). Сервер, синхронизация которого происходит с использованием пакетов, помеченных как Stratum n, принадлежит к следующему, более низкому, уровню, и его пакеты помечаются как Stratum n+1. Серверы одного и того же уровня могут обмениваться пакетами друг с другом, но относятся к одному и тому же уровню, т. е. на один уровень ниже лучшего коррелятора, с которым они могут синхронизироваться. Уровень назначения 16 (Stratum 16) используется для обозначения того, что сервер в настоящий момент не выполняет синхронизацию с каким-либо надежным источником времени.

Обратите внимание, что по умолчанию для систем, находящихся уровнем ниже, клиенты NTP работают как серверы.

16.1.1.1. Краткий обзор уровней NTP

Уровень 0, Stratum 0

- атомные часы и их сигналы, передаваемые по радио и GPS;
- система глобального позиционирования GPS (Global Positioning System);
- системы мобильной телефонной связи;
- низкочастотное радиовещание: станция WWVB (Колорадо, США), станции JJY-40 и JJY-60 (Япония), станция DCF77 (Германия) и станция MSF (Великобритания).

Вышеуказанные сигналы можно принимать с помощью специально предназначенных для этого устройств, обычно подключенных через интерфейс RS-232 к системе, используемой как сервер времени организации или сайтов.

Уровень 1, Stratum 1

Компьютер с подключенными к нему радио-часами, часами GPS или атомными часами.

Уровень 2, Stratum 2

Читает данные с уровня 1, служит сервером для нижнего уровня

Уровень 3, Stratum 3

Читает данные с уровня 2, служит сервером для нижнего уровня

Уровень n+1, Stratum n+1

Читает данные с уровня n, служит сервером для нижнего уровня

Уровень 15, Stratum 15

Читает данные с уровня 14; это самый низкий уровень.

Процесс снижается до уровня 15, являющегося нижайшим действительным уровнем. Метка Stratum 16 используется для обозначения статуса «без синхронизации».

16.1.2. UTC, часовые пояса и переход на летнее время

Поскольку NTP функционирует полностью на основе UTC (всемирное координированное время, Coordinated Universal Time), часовые пояса и переход на летнее время применяются в системах локально. Файл /etc/localtime является копией или символьной ссылкой на файл информации о часовом поясе из каталога /usr/share/zoneinfo. Системные часы могут отсчитывать местное время или UTC, что указывается в третьей строке файла /etc/adjtime, которая может иметь значение либо «LOCAL» либо «UTC» для обозначения того, как именно были настроены системные часы (т. е. часы реального времени, RTC). Как правило, рекомендуется настраивать часы реального времени на UTC, чтобы избежать различных проблем, связанных с переходом на летнее время.

16.1.3. Файл смещения

В файле смещения (drift file) обычно хранится значение смещения частоты между системными часами, работающими с номинальной частотой, и частотой, которая требуется для того, чтобы часы оставались синхронизированными с UTC. При наличии этого значения в файле смещения оно читается во время старта системы и используется для коррекции источника времени. Использование файла смещения сокращает время, требуемое для получения стабильного и точного времени. Расчет значения и соответствующая замена файла смещения производится один раз в час службой ntpd. Файл смещения заменяется, а не обновляется, поэтому важно, чтобы у службы ntpd имелись права на запись в соответствующий каталог.

16.1.4. Возможности аутентификации для NTP

В NTPv4 была добавлена поддержка для архитектуры системы безопасности Autokey (автоключ), основанной на открытом асимметричным шифровании, и в тоже время по-прежнему поддерживающей шифрование с симметричным ключом. Протокол Autokey описан в документе RFC 5906. К сожалению, позже было обнаружено, что у протокола имеются серьезные проблемы безопасности, и было рекомендовано использовать симметричный ключ. На странице руководства ntp_auth(5) описываются параметры и команды аутентификации для ntpd.

Злоумышленник может попытаться прервать выполнение службы с помощью отправки пакетов с неправильной информацией о времени. В системах, использующих открытый пул серверов NTP, риск снижается наличием нескольких серверов NTP в списке общедоступных серверов файла /etc/ntp.conf. Если только один источник времени будет скомпрометирован, ntpd проигнорирует этот источник. Администратор должен выполнить оценку рисков и обдумать влияние неточного времени на ресурсы подотчетной организации. При наличии внутренних источников времени нужно обдумать шаги, которые необходимо выполнить для защиты сети, по которой распространяются пакеты NTP. Если в итоге

будет принято решение, что риски приемлемы, аутентификацию можно не использовать.

Для широковещательного и группового режимов по умолчанию требуется аутентификация. При использовании доверенной сети аутентификацию можно отключить при помощи директивы disable auth в файле ntp.conf. Настроить аутентификацию можно при помощи симметричных ключей SHA1 или MD5 или же с помощью открытого асимметричного шифрования по схеме автоключа. Их описания содержатся на страницах руководств ntp_auth(8) и ntp-keygen(8).

16.1.5. Настройка симметричной аутентификации с использованием ключа

Для настройки симметричной аутентификации с использованием ключа добавьте следующий параметр после команды server или peer:

```
key <число>
```

Здесь <число> может принимать значения от 1 до 65534 включительно. Этот параметр включает использование в пакетах кода проверки подлинности сообщения MAC и используется с командами peer, server, broadcast и manycastclient. В файле /etc/ntp.conf этот параметр используется следующим образом:

```
server 192.168.1.1 key 10
broadcast 192.168.1.255 key 20
manycastclient 239.255.254.254 key 30
```

16.1.6. Конфигурационный файл NTP

Демон ntpd читает параметры конфигурационного файла при запуске системы или во время перезапуска службы. Местоположение файла по умолчанию — /etc/ntp.conf. Просмотреть файл можно с помощью команды cat:

```
$ cat /etc/ntp.conf
```

Команды конфигурации кратко описываются в разделе «Настройка NTP» и более подробно — на странице руководства ntp.conf(5)

Ниже объясняется содержимое конфигурационного файла по умолчанию.

Раздел файла смещения (driftfile)

Здесь указывается путь до файла смещения. Значение по умолчанию — driftfile /var/lib/ntp/drift.

При смене местоположения файла убедитесь, что служба ntpd имеет права на запись в этот каталог. Файл содержит только значения, используемые для настройки частоты системных часов после каждого старта системы или старта службы. Подробности см. в подразделе «Файл смещения».

Раздел параметров контроля доступа

Следующая запись настраивает параметры контроля доступа по умолчанию:

```
restrict default nomodify notrap nopeer noquery
```

- nomodify запрещает внесение изменений в параметры;
- notrap запрещает ловушки протокола контроля сообщений ntpdc;
- nopeer запрещает создание связей между узлами одноранговой сети;

- noquery запрещает ответ на запросы ntpq и ntpdc, но разрешает запросы времени.

Примечание. Запросы ntpq и ntpdc могут быть использованы злоумышленниками в атаках с лавинообразным умножением данных (amplification attacks), поэтому не следует удалять параметр noquery из параметров команды restrict по умолчанию в общедоступных системах. Подробности см. по ссылке: https://access.redhat.com/security/cve/CVE-2013-5211.

Адреса в диапазоне 127.0.0.0/8 часто требуются разным процессам или приложениям. Как строка restrict default предотвращает доступ ко всему, не разрешенному явно, так и доступ к стандартному адресу петли (loopback) для IPv4 и IPv6 разрешается в следующих строках:

```
# the administrative functions.
restrict 127.0.0.1
restrict ::1
```

Явно требуемые приложениями адреса можно добавить ниже.

Хосты из локальной сети запрещаются записью restrict default, описанной выше. Чтобы изменить это поведение, например, разрешить только запросы времени и статистики из сети 192.0.2.0/24, требуется запись следующего формата:

```
restrict 192.0.2.0 mask 255.255.255.0 nomodify notrap nopeer
```

Чтобы разрешить полный доступ для конкретного хоста, например, 192.0.2.250/32, требуется запись следующего формата:

```
restrict 192.0.2.250
```

Если маска не указана явно, применяется маска 255.255.255.255.

Команды restrict объясняются на странице руководства ntp_acc(5).

Раздел общедоступных серверов (public servers)

По умолчанию файл ntp.conf содержит две записи для общедоступных серверов:

```
server ntp.rosalinux.ru iburst
server ntp2.rosalinux.ru iburst
```

Раздел многоадресных широковещательных серверов (broadcast multicast servers)

По умолчанию в файле ntp.conf содержится несколько закомментированных примеров. В основном, эти примеры не требуют разъяснений. Объяснение конкретных команд см. в разделе «Настройка NTP». При необходимости добавляйте нужные команды непосредственно под примерами.

Примечание. Когда клиентская программа DHCP, dhclient, получает список серверов NTP от сервера DHCP, они добавляются в ntp.conf, после чего служба NTP перезапускается. Для отключения этого поведения добавьте в файл /etc/sysconfig/network запись PEERNTP=no.

16.1.7. Файл sysconfig службы ntpd

Файл читается начальным сценарием ntpd при запуске службы. Содержимое файла по умолчанию:

```
# Command line options for ntpd
OPTIONS="-g"
```

Параметр -g разрешает ntpd проигнорировать предел смещения в 1000 секунд и попробовать синхронизировать время даже в том случае, если смещение составляет более 1000 секунд, но только при загрузке системы. Без этого параметра, если смещение составляет больше 1000 секунд, ntpd завершит работу. ntpd также завершит работу даже и при наличии параметра -g, если после загрузки системы был произведен перезапуск службы NTP, и смещение составляет больше 1000 секунд.

16.1.8. Установка демона NTP (ntpd)

Сервер сетевого времени NTP реализован в виде демона (или службы) ntpd, который содержится в пакете ntp.

Чтобы установить ntpd, выполните следующую команду с привилегиями суперпользователя root:

```
# yum install ntp
```

Чтобы включить запуск ntpd при загрузке системы, выполните:

```
# systemctl enable ntpd
```

16.1.9. Проверка статуса NTP

Чтобы проверить, выполняется ли служба ntpd и настроена ли она на автоматический запуск при старте системы, выполните следующую команду:

```
$ systemctl status ntpd
```

Чтобы получить краткую информационную сводку от службы ntpd, выполните:

```
$ ntpstat
unsynchronised
  time server re-starting
   polling server every 64 s

$ ntpstat
synchronised to NTP server (10.5.26.10) at stratum 2
   time correct to within 52 ms
   polling server every 1024 s
```

16.1.10. Настройка NTP

Чтобы изменить параметры по умолчанию службы NTP в файле /etc/ntp.conf, используйте текстовый редактор, запущенный с правами root. Файл устанавливается вместе со службой ntpd. На странице руководства ntp.conf(5) описаны параметры, которые можно использовать в конфигурационном файле, помимо команд доступа и ограничения скорости ответа, которые объясняются на странице руководства ntp_acc(5).

16.1.10.1. Настройка контроля доступа к службе NTP

Для запрета или контроля доступа к запущенной в системе службе NTP используйте команду restrict в файле ntp.conf. См. закомментированный пример:

```
# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

Команда restrict имеет следующий вид:

```
restrict <параметр>
```

Здесь <параметр> — это один или несколько следующих:

- ignore — игнорируются все пакеты, включая запросы ntpq и ntpdc;
- kod — для снижения количества нежелательных запросов необходимо послать пакет «Kiss-o'-death»;
- limited — не отвечать на запросы о предоставлении службы времени, если пакет нарушает значения ограничения скорости, установленные по умолчанию, или же указанные командой discard. На запросы ntpq и ntpdc этот параметр не влияет;
- lowpriotrap — ловушкам, установленным хостами, отвечающими указанному шаблону, присваивается низкий приоритет.
- nomodify — запрещает любые попытки изменения параметров;
- noquery — запрещает ответы на запросы ntpq и ntpdc, но не на запросы времени;
- nopeer — предотвращает создание связей между узлами одноранговой сети;
- noserve — сбрасывает все пакеты, кроме запросов ntpq и ntpdc;
- notrap — запрещает создание ловушек для управляющих сообщений ntpdc;
- notrust — сбрасывает все пакеты с аутентификацией без шифрования;
- ntpport — изменяет алгоритм сравнения таким образом, что ограничение применяется только если исходный порт является стандартным портом NTP UDP — 123;
- version — сбрасывает все пакеты, не совпадающие с текущей версией NTP.

Чтобы доступ с ограничением скорости трафика запрещал ответ на все запросы, соответствующая команда restrict должна иметь параметр limited. Если ntpd должен отвечать пакетом KoD, команда restrict должна иметь оба параметра — и limited, и kod.

Запросы ntpq и ntpdc могут использованы в атаках с лавинообразным умножением данных («amplification attack», подробности см.: https://access.redhat.com/security/cve/CVE-2013-5211), поэтому не удаляйте параметр noquery из параметров по умолчанию для команды restrict в общедоступных системах.

16.1.10.2. Настройка доступа с ограничением интенсивности трафика для службы NTP

Чтобы включить ограничение по скорости трафика для доступа к службе NTP, работающей в системе, добавьте параметр limited для команды restrict, как это объяснялось в разделе «Настройка контроля доступа к службе NTP». Если этот параметр по умолчанию по каким-либо причинам не используется, применяйте команду discard, как объясняется ниже.

Команда discard имеет следующий вид:

```
discard [average <значение>] [minimum <значение>] [monitor <зна-
чение>]
```

Здесь:

- average — указывает минимальный средний разрешенный интервал между пакетами, принимает аргумент в log2 секунд. Значение по умолчанию — 3 (23 равно 8 се-

кундам);

- minimum — указывает минимальный разрешенный интервал между пакетами, принимает аргумент в log2 секунд. Значение по умолчанию — 1 (21 равно 2 секундам);
- monitor — указывает возможность сброса для пакетов при превышении разрешенного ограничения. Значение по умолчанию — 3000 секунд. Этот параметр предназначается для серверов, получающих 1000 или более запросов в секунду.

Примеры команды discard:

```
discard average 4
discard average 4 minimum 2
```

16.1.10.3. Добавление адреса узла одноранговой сети

Чтобы добавить адрес узла одноранговой сети, то есть адрес сервера, на котором выполняется служба NTP и который расположен на том же уровне (stratum), в файле ntp.conf используется команда peer:

```
peer <адрес>
```

Здесь <адрес> — это одиночный адрес IP или имя, разрешаемое DNS. Адрес должен принадлежать системе, про которую известно, что она находится на одном уровне NTP (stratum) с вашей системой. У каждого узла одноранговой сети должен быть по крайней мере один источник времени, отличный от источников времени другого узла. Обычно узлами одноранговой сети являются системы в рамках одного и того же административного управления.

16.1.10.4. Добавление адреса сервера

Чтобы добавить адрес сервера, на котором выполняется служба NTYP и который находится на более высоком уровне (stratum) NTP, в файле ntp.conf используется команда server:

```
server <адрес>
```

Здесь <адрес> — это одиночный адрес IP или имя, разрешаемое DNS. Это адрес удаленного запрашиваемого сервера или сервера местных справочных часов, с которого нужно получать пакеты.

16.1.10.5. Добавление адреса широковещательного или многоадресного сервера

Чтобы добавить широковещательный или многоадресный адрес назначения, то есть адрес, на который нужно посылать широковещательные или многоадресные пакеты NTP, в файле ntp.conf используется команда broadcast.

Широковещательные и многоадресные режимы по умолчанию требуют аутентификации. См. подраздел «Возможности аутентификации NTP».

Команда broadcast имеет следующий вид:

```
broadcast <адрес>
```

Здесь <адрес> — это широковещательный или множественный IP-адрес, на который должны посылаться пакеты.

Эта команда превращает систему в широковещательный сервер NTP. Широковеща-

тельный адрес предполагает адрес IPv4 255.255.255.255. по умолчанию; маршрутизаторы не передают широковещательных сообщений. Широковещательный адрес должен быть адресом IPv4 класса D или адресом IPv6. Администрация адресного пространства интернет (IANA) присвоила службе NTP адреса многоадресной рассылки IPv4 224.0.1.1 и IPv6 FF05::101 (внутрисайтовый). Также можно использовать многовещательные адреса IPv4 административного назначения, как опоясано в документе RFC 2365 (https://www.rfc-editor.org/info/rfc2365).

16.1.10.6. Добавление клиентского адреса manycast

Чтобы добавить клиентский адрес manycast, то есть настроить адрес многоадресного вещания так, чтобы он мог использоваться для обнаружения серверов NTP, в файле ntp.conf используется команда manycastclient:

```
manycastclient <адрес>
```

Здесь <адрес> — это IP-адрес многоадресного вещания, с которого нужно получать пакеты. На этот адрес клиент посылает запрос, среди ответов выбираются лучшие серверы, другие игнорируются. Затем соединение NTP использует одноадресные связи, как если бы обнаруженные серверы NTP были указаны в файле ntp.conf.

Эта команда превращает систему в клиент NTP. Система может быть одновременно как клиентом, так и сервером.

16.1.10.7. Добавление широковещательного клиентского адреса

Чтоб добавить широковещательный клиентский адрес, то есть настроить широковещательный адрес так, чтобы он отслеживался широковещательными пакетами NTP, в файле ntp.conf используется команда broadcastclient:

```
broadcastclient
```

Команда активирует получение широковещательных сообщений. По умолчанию требуется аутентификация. См. подраздел «Возможности аутентификации для NTP»

Эта команда превращает систему в клиент NTP. Система может быть одновременно как клиентом, так и сервером.

16.1.10.8. Добавление серверного адреса manycast

Чтоб добавить серверный адрес manycast, то есть настроить адрес так, чтобы клиенты могли обнаруживать сервер с помощью многоадресных пакетов NTP, в файле ntp.conf используется команда manycastserver:

```
manycastserver <адрес>
```

Команда разрешает рассылку многоадресных сообщений, где <адрес> — это адрес, на который нужно посылать сообщения. Чтобы избежать перебоев в работе службы, для этой команды необходима аутентификация.

Эта команда превращает систему в сервер NTP. Система может быть одновременно как клиентом, так и сервером.

16.1.10.9. Добавление многоадресного адреса клиента

Чтобы добавить многоадресный клиентский адрес, то есть настроить многоадресный адрес так, чтобы он отслеживался многоадресным пакетами NTP, в файле ntp.conf

используется команда multicastclient:

```
multicastclient <адрес>
```

Команда разрешает получение многоадресных сообщений, где <адрес> — это адрес подписки. Чтобы избежать перебоев в работе службы, для этой команды необходима аутентификация.

Эта команда превращает систему в клиент NTP. Система может быть одновременно как клиентом, так и сервером.

16.1.10.10. Параметр burst

Использование параметра burst для общедоступного сервера считается некорректным. Не используйте этот параметр на общедоступных серверах NTP. Используйте ее только для приложений в рамках своей организации.

Для повышения общего качества статистики смещения времени добавьте следующий параметр после команды server:

```
burst
```

В каждый интервал опроса при ответе сервера система будет посылать серию до восьми пакетов вместо обычного одного пакета.

16.1.10.11. Параметр iburst

Для улучшения времени, используемого для начальной синхронизации, добавьте следующий параметр после команды server:

```
Iburst
```

Если сервер недоступен, будет послана серия из 8 пакетов вместо обычного одного. Промежуток между пакетами обычно составляет 2 секунды; тем не менее, промежуток между первым и вторым пакетами можно изменить с помощью команды calldelay для получения дополнительного времени для завершения звонка модема или ISDN. Используется вместе с командой server для сокращения времени начальной синхронизации. Iburst является параметром по умолчанию в конфигурационном файле.

Приведем пример создания самого простого сервера NTP, с которого ваши клиенты смогут получать данные для синхронизации времени. Эта инструкция будет полезна в случае, если у вас есть закрытая сеть без выхода в интернет.

Для настройки сервера точного времени ntpd выполните следующие действия:

1) Установить сервис ntp, если он еще не установлен, следующей командой:
   ```
   yum install ntp
   ```
2) Чтобы использовать ntpd в качестве службы сетевого времени по умолчанию, необходимо остановить и отключить демон chronyd. Выполните следующие команды:
   ```
   # systemctl stop chronyd
   # systemctl disable chronyd
   ```
3) Для настройки автоматического запуска демона при загрузке системы используйте специальную команду:
   ```
   systemctl enable ntpd
   ```
4) Отредактируйте конфигурационный файл сервера /etc/ntp.conf. Он должен содержать как минимум следующие данные:

```
driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/
#каталог для сбора статистики
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
server 127.127.1.0
fudge 127.127.1.0 stratum 0
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
restrict 127.0.0.1
restrict <ваша сеть> mask <маска вашей сети> nomodify notrap
```

5) Перезагрузите демон NTP:

```
systemctl restart ntpd
```

6) Добавьте ntp в автозагрузку:

```
systemctl enable ntpd
```

Расшифровка основных параметров

- driftfile — указывает файл для хранения информации о частоте смещения времени. В этом файле хранится значение, получаемое в результате предшествующих корректировок времени. Если внешние NTP-серверы по той или иной причине становятся недоступными, значение будет взятого из него;
- statsdir — каталог для сбора статистики работы сервиса;
- server — укажите, если ваш сервер будет сам обновлять свое время с некоторого внешнего сервера. Если верхних серверов NTP нет, то указывается 127.127.1.0. В этом случае будет использоваться локальное время ОС;
- restrict — ограничивает работу сервиса в определенной подсети, например:

  ```
  restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
  ```

  Сервер NTP будет отвечать на запросы только из подсети 192.168.1.0/24. Крайне рекомендуется использовать данный параметр, чтобы ограничить нагрузку на сервер.

16.1.10.12. Дополнительные источники информации

Дополнительную информацию о NTP и ntpd можно найти в следующих руководствах:

- страница руководства ntpd(8) — подробное описание ntpd, включая параметры командной строки;
- страница руководства ntp.conf(5) — содержит информацию о том, как настраивать связи с серверами и узлами одноранговой сети;
- страница руководства ntpq(8) — описывается утилита запросов NTP, используемая для выполнения запросов и отслеживания сервера NTP;
- страница руководства ntpdc(8) — описывается утилита службы ntpd, используемая

для запросов и смены статуса ntpd;

- страница руководства ntp_auth(5) — описывается параметры ntpd, команды и управление ключами аутентификации для ntpd;
- страница руководства ntp_keygen(8) — описывается создание открытых и частных ключей для ntpd;
- страница руководства ntp_acc(5) — описывается контроль доступа с использованием команды restrict;
- страница руководства ntp_mon(5) — описываются возможности мониторинга для сбора статистики;
- страница руководства ntp_clock(5) — описываются команды для настройки опорной частоты;
- страница руководства ntp_misc(5) — описываются дополнительные параметры;
- страница руководства ntp_decode(5) — список слов состояния, сообщений о событиях и кодах ошибок, используемых для отчетов и наблюдений за ntpd;
- страница руководства ntpstat(8) — описывается утилита, используемая для получения статуса синхронизации демона NTP, выполняемого на локальной машине;
- страница руководства ntptime(8) — описывается утилита для чтения и установки переменных времени в ядре;
- страница руководства tickadj(8) — описывается утилита для чтения (и возможной установки) тактовой длины.

16.2. Настойка сервера DHCP

Данная инструкция не претендует на полное описание всех возможностей работы сервиса dhcp, а предлагает простой способ настройки сервера динамической конфигурации сети для быстрого старта.

1) Для создания сервера dhcp необходимо установить соответствующую службу:
   ```
   yum install dhcp
   ```
2) Для настройки автоматического запуска демона при загрузке системы используйте специальную команду:
   ```
   systemctl enable dhcpd
   ```
3) Необходимо настроить один из интерфейсов сервера на статический адрес из той подсети, которую будет раздавать клиентам, иначе демон не будет работать корректно.

Для настройки сервиса нужно сначала скопировать файл с типовой конфигурацией сервиса /usr/share/doc/dhcp-<версия>/dhcpd.conf.sample в каталог /etc/dhcp/, переименовав его в файл dhcpd.conf:

```
cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example
/etc/dhcp/dhcpd.conf
```

После этого следует отредактировать файл /etc/dhcpd.conf, указав в нем нужные параметры:

```
mcedit /etc/dhcp/dhcpd.conf
```

Приведите файл к следующему виду:

```
option domain-name "test.dom";
option domain-name-servers 192.168.10.1;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;

subnet 192.168.10.0 netmask 255.255.255.0 {
  range 192.168.10.10 192.168.10.200;
  option routers 192.168.10.254;
  option broadcast-address 192.168.10.255;
}
```

В данном примере предполагается, что станции, получающие сетевые настройки, работают в домене test.dom с сервером DNS 192.168.10.1 и шлюзом по умолчанию 192.168.10.254 и получают IP-адреса в промежутке от 192.168.10.10 до 192.168.10.200 с маской подсети 255.255.255.0.

Описание параметров

- `option domain-name` — определяет имя домена. Глобальный параметр. По умолчанию для всех подсетей;
- `option domain-name-servers` — определяет список адресов серверов DNS через запятую. Глобальный параметр. По умолчанию для всех подсетей;
- `default-lease-time` — время аренды по умолчанию;
- `max-lease-time` — определяет максимально допустимое время аренды. Независимо от длительности аренды, фигурирующей в запросе клиента, этот срок не может превышать значение, заданное данным параметром;
- `authoritative` — означает, что в вашей сети данный сервер является ответственным за выдачу сетевых адресов;
- `log-facility` — определяет направление потока логов;
- `subnet` — основной логический блок конфигурации. Он определяет настройки для конкретной сети. В том числе в нем можно менять глобальные параметры, такие как domain-name, domain-name-servers и др.;
- `range` — диапазон IP-адресов, доступный для аренды;
- `option routers` — адрес маршрутизатора по умолчанию;
- `option broadcast-address` — адрес для широковещательной рассылки.

После сохранения изменений в конфигурационном файле необходимо перезагрузить сервис dhcpd:

```
/etc/init.d/dhcpd restart
```

16.3. Веб-сервер Apache

Веб-сервер, поставляемый в составе ОС РОСА «КОБАЛЬТ», — это Apache HTTP Server версии 2.4.

16.3.1. Особенности версии Apache 2.4 в ОС РОСА «КОБАЛЬТ»

Отдельный каталог /tmp

Для повышения уровня защищенности системы юнит systemd выполняет демон httpd с использованием частного каталога /tmp, отдельно от системного каталога /tmp.

Схема конфигурации по пакетам

Файлы конфигурации, с помощью которых загружаются модули, теперь располагаются в каталоге /etc/httpd/conf.modules.d/. Пакеты, предоставляющие дополнительные загружаемые модули для httpd, например, php, разместят файлы в этом каталоге. Для включения таких файлов в каталог /etc/httpd/conf.modules.d/ в файле /etc/httpd/conf/httpd.conf существует директива Include, следующая перед главным разделом. Это означает, что любые файлы конфигураций, располагающиеся в каталоге /conf.modules.d, обрабатываются до начала обработки основной информации файла httpd.conf. Директива IncludeOptional для файлов из каталога /etc/httpd/conf.d/ помещается в конце файла httpd.conf. Это означает, что файлы из каталога /etc/httpd/conf.d/ теперь обрабатываются после обработки основной информации из httpd.conf.

Некоторые конфигурационные файлы предоставляются пакетом httpd:

- /etc/httpd/conf.d/autoindex.conf — параметры индексации каталога mod_autoindex;
- /etc/httpd/conf.d/userdir.conf — параметры доступа в пользовательские каталоги, например http://test.dom/~username/. Такой доступ по умолчанию отключен из соображений безопасности;
- /etc/httpd/conf.d/welcome.conf — как и в предыдущих релизах, этот файл отвечает за страничку приветствия, показываемую по адресу http://localhost/ в отсутствие другой информации.

Конфигурация по умолчанию

По умолчанию файл httpd.conf содержит минимальную конфигурацию. Многие обычные параметры, такие, как Timeout или KeepAlive, не настраиваются явно в конфигурации по умолчанию; вместо них по умолчанию используются встроенные параметры, см. подраздел «Устанавливаемая документация».

Модель обработки

В ОС РОСА «КОБАЛЬТ» используется только один бинарный файл httpd, а три модели MPM доступны в виде загружаемых модулей: worker, prefork (по умолчанию) и event. Отредактируйте файл /etc/httpd/conf.modules.d/00-mpm.conf согласно требованиям конкретной системы, добавляя и убирая символ комментария # так, чтобы загружался только один модуль MPM.

Аутентификация, авторизация и контроль доступа

Для управления аутентификацией, авторизацией и контролем доступа используется синтаксис Require, значительно отличающийся от директив Order, Deny и Allow. По-

дробности см. в документе организации Apache: http://httpd.apache.org/docs/2.4/howto/auth.html.

suexec

В целях повышения уровня безопасности системы исполняемый файл suexec больше не устанавливается как if пользователем root. Вместо этого у него появился набор битов, устанавливающих права на данной файловой системе для более строгого набора прав доступа. В дополнение к этому изменению, бинарный файл suexec больше не использует файл журнала /var/log/httpd/suexec.log. Теперь сообщения журнала посылаются в syslog; по умолчанию они появляются в файле журнала /var/log/secure.

16.3.2. Выполнение службы httpd

В данном подразделе описывается, как запустить, остановить, перезапустить и проверить текущий статус сервера Apache HTTP. Прежде чем использовать службу httpd, убедитесь, что в системе установлен httpd. Это можно сделать, выполнив следующую команду:

```
# yum install httpd
```

16.3.2.1. Запуск службы

Чтобы запустить службу httpd, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl start httpd.service
```

Для автоматического запуска службы при загрузке системы выполите:

```
# systemctl enable httpd.service
```

Примечание. Если сервер Apache HTTP работает как защищенный сервер, при использовании зашифрованного закрытого ключа SSL после загрузки компьютера потребуется ввести пароль.

16.3.2.2. Остановка службы

Чтобы остановить выполняющуюся службу httpd, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl stop httpd.service
```

Чтобы предотвратить автоматический запуск службы при загрузке системы, выполните:

```
# systemctl disable httpd.service
```

16.3.2.3. Перезапуск службы

Существуют три разных способа перезапустить выполняющуюся службу httpd.

1) Чтобы полностью перезапустить службу, выполните следующую команду:

```
# systemctl restart httpd.service
```

Это действие останавливает выполняющуюся службу httpd и немедленно запускает ее снова. Эта команда используется после установки или удаления динамически загружаемого модуля, например, PHP.

2) Чтобы просто перезагрузить конфигурацию, выполните:

```
# systemctl reload httpd.service
```

Это действие заставит работающую службу httpd перезагрузить файл конфигурации. Все запросы, обрабатываемые в это время, будут прерваны, что может вызвать показ сообщения об ошибке в браузере клиента или неполную загрузку страницы.

3) Для перезагрузки конфигурации, не отражающейся на активных запросах, выполните:

```
# apachectl graceful
```

Это действие заставит работающую службу httpd перезагрузить файл конфигурации. Все запросы, обрабатываемые в это время, будут использовать старую конфигурацию.

16.3.2.4. Проверка статуса службы

Чтобы проверить, работает ли служба httpd, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl is-active httpd.service
```

16.3.2.5. Редактирование файлов конфигурации

При запуске служба httpd по умолчанию читает конфигурационные файлы из следующих местоположений в системе:

- /etc/httpd/conf/httpd.conf — главный файл;
- /etc/httpd/conf.d/ — вспомогательный каталог для конфигурационных файлов, включаемых в главный файл.

Хотя параметры по умолчанию подходят для большинства ситуаций, желательно познакомиться с некоторыми из наиболее важных параметров конфигурации. Обратите внимание: чтобы изменения конфигурации вступили в силу, сервер сначала нужно перегрузить.

Чтобы проверить конфигурацию на наличие ошибок, выполните следующую команду с привилегиями суперпользователя root:

```
# apachectl configtest
```

Чтобы облегчить процесс устранения ошибок, рекомендуется сделать резервную копию исходного файла перед его изменением.

16.3.2.6. Работа с модулями

Являясь модульным приложением, служба httpd поставляется вместе с некоторым числом модулей, которые при необходимости можно динамически загружать и выгружать в рабочем режиме. В ОС РОСА «КОБАЛЬТ» эти модули располагаются в каталоге /usr/lib64/httpd/modules/.

Загрузка модулей

Чтобы загрузить модуль, используйте директиву LoadModule в одном из конфигурационных файлов в каталоге /etc/httpd/conf.modules.d. Обратите внимание, что модули, предоставляемые в отдельных пакетах, часто имеют свой собственный конфигурационный файл в каталоге /etc/httpd/conf.d/.

Пример: загрузка динамического разделяемого модуля mod_ssl

LoadModule ssl_module modules/mod_ssl.so

Дождавшись выполнения команды, перезагрузите сервер для обновления конфигурации.

Написание модулей

Администраторам, желающим написать свой собственный динамический разделяемый модуль, нужно убедиться в том, что в системе установлен пакет httpd-devel. Для этого выполните следующую команду с привилегиями суперпользователя root:

```
# yum install httpd-devel
```

В этом пакете содержатся файлы include, файлы заголовков и утилита APache eXtenSion (apxs), необходимая для компиляции модуля.

После написания модуля соберите его с помощью следующей команды:

```
# apxs -i -a -c module_name.c
```

Если результат сборки был удачен, модуль можно загружать точно так же, как и любой другой модуль, идущий в составе сервера Apache HTTP.

16.3.3. Настройка межсетевого экрана для разрешения трафика HTTP и HTTPS

ОС РОСА «КОБАЛЬТ» по умолчанию не разрешает трафик HTTP и HTTPS. Чтобы дать возможность системе работать как веб-сервер, убедитесь, что службы, поддерживаемые firewalld, разрешают пропуск трафика HTTP и HTTPS сквозь межсетевой экран.

Чтобы включить HTTP в консоли, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --add-service http
```

Чтобы включить HTTPS в консоли, выполните:

```
# firewall-cmd --add-service https
```

Обратите внимание, что эти изменения будут действовать только до следующей перезагрузки системы. Чтобы сделать это изменение постоянным, повторно выполните команду с параметром `--permanent`.

16.3.3.1. Проверка сетевого доступа для входящего трафика HTTPS и HTTPS

Чтобы проверить, какие службы разрешены в межсетевом экране, выполните следующую команду с привилегиями суперпользователя root:

```
# firewall-cmd --list-all
```

В выводе команды вы должны увидеть строки, разрешающие входящие соединения для протоколов http и https.

16.3.4. Некоторые полезные параметры файла /etc/httpd/conf/httpd.conf

User http

По соображениям безопасности при запуске сервера Apache от имени суперпользователя (напрямую или через скрипт инициализации) происходит смена идентификатора пользователя (UID), от имени которого выполняется процесс сервера. По умолчанию используется пользователь http, который создается при установке и не имеет привилегированных полномочий в системе.

Listen 80

Это порт, через который Apache принимает входящие соединения. Если сервер имеет выход в интернет через маршрутизатор, необходимо будет настроить перенаправление этого порта.

Если Apache используется для разработки и тестирования, лучше разрешить только локальный доступ к нему. Для этого укажите `Listen 127.0.0.1:80`.

ServerAdmin you@test.dom

Адрес электронной почты администратора, который будет выводиться, например, на странице ошибки Apache.

DocumentRoot "/srv/http"

Это корневой каталог Apache, в котором можно разместить ваши веб-страницы.

Измените его, если нужно, но не забудьте также поменять путь в директиве `<Directory "/srv/http">` на новое расположение DocumentRoot, иначе вы, скорее всего, получите сообщение об ошибке «403 Error» (недостаточно полномочий) при попытке получить доступ к новому корневому каталогу Apache. Также не забудьте изменить строку Require all denied на Require all granted, иначе снова получите ошибку 403 Error. Помните, что каталог DocumentRoot и его родительские папки должны иметь разрешения на запуск для всех (можно установить командой `chmod o+x /путь/у/DocumentRoot`), в противном случае вы получите ошибку 403 Error.

AllowOverride None

Запрещает переопределение настроек. Если в секции `<Directory>` указана эта директива, Apache будет полностью игнорировать настройки в файле .htaccess. Обратите внимание, что такая настройка для Apache 2.4 является настройкой по умолчанию, поэтому если вы планируете использовать .htaccess, вам необходимо дать соответствующие разрешения. Если вы собираетесь включить модуль mod_rewrite или использовать настройки в .htaccess, вы можете определить, какие из директив, объявленных в этих файлах, могут перезаписывать конфигурацию сервера. Для получения дополнительной информации обратитесь к документации Apache: http://httpd.apache.org/docs/current/mod/core.html#allowoverride.

Дополнительные настройки можно найти в /etc/httpd/conf/extra/httpd-default.conf.

Чтобы полностью отключить вывод версии Apache в генерируемых сервером страницах, добавьте директиву `ServerSignature Off`. Чтобы подавить вывод такой информации, как версии Apache и PHP, добавьте `ServerTokens Prod`.

16.3.5. Пользовательские каталоги

По умолчанию доступ к каталогам пользователей возможен по адресу http://localhost/~"user"/, который показывает содержимое каталога ~/public_html (его имя и расположение задаются в файле /etc/httpd/conf/extra/httpd-userdir.conf).

Если вы не хотите, чтобы пользовательские каталоги были доступны через web, закомментируйте следующую строку в /etc/httpd/conf/httpd.conf:

```
Include conf/extra/httpd-userdir.conf
```

Убедитесь, что права доступа к вашему домашнему каталогу и ~/public_html позволяют получать доступ к файлам в них всем пользователям:

```
$ chmod o+x ~
$ chmod o+x ~/public_html
$ chmod -R o+r ~/public_html
```

Однако с точки зрения безопасности вышеприведенное решение ненадежно. Правильнее поступить следующим образом:

1) Добавьте пользователя http в группу, которой принадлежит ваша домашняя папка. Например, если ваша домашняя папка и все ее подкаталоги принадлежат группе piter, можно проделать следующее:

   ```
   # usermod -aG piter http
   ```

   или

   ```
   # gpasswd -a http piter
   ```

2) Назначьте права на чтение и исполнение для каталогов ~/, ~/public_html и, рекурсивно, на остальные подкаталоги для ~/public_html для членов группы (в нашем примере для членов группы piter). Следуйте нижеприведенному образцу:

   ```
   $ chmod g+xr-w /home/yourusername
   $ chmod -R g+xr-w /home/yourusername/public_html
   ```

Примечание. В результате только пользователь http и все потенциальные пользователи группы piter будут иметь разделяемый доступ к вашему домашнему каталогу.

3) Перезапустите службу httpd, чтобы изменения вступили в силу.

16.3.6. TLS/SSL

Для использования TLS/SSL необходимо установить openssl.

Создайте закрытый ключ и запрос на получение сертификата (CSR). Также вы можете создать самоподписанный сертификат:

Примечание. Вы можете настроить длину ключа в битах (`rsa_keygen_bits:2048`). Также вы можете убрать опцию `-sha256` для использования SHA-1 вместо SHA-2 или изменить время его действия в днях (`-days 365`).

```
# cd /etc/httpd/conf
# openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048
-out server.key
# chmod 600 server.key
# openssl req -new -sha256 -key server.key -out server.csr
# openssl x509 -req -days 365 -in server.csr -signkey server.key
-out server.crt
Теперь раскомментируйте следующие строки в
/etc/httpd/conf/httpd.conf:

LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
Include conf/extra/httpd-ssl.conf
```

Перезапустите службу httpd.service, чтобы изменения вступили в силу.

16.3.7. Виртуальные хосты

Примечание. Необходимо добавить отдельную секцию `<VirtualHost dommainame:443>` для поддержки SSL на виртуальном хосте. Пример файла можно посмотреть ниже.

Если вы хотите, чтобы Apache обслуживал не один, а несколько хостов, раскомментируйте следующую строку в файле /etc/httpd/conf/httpd.conf:

```
Include conf/extra/httpd-vhosts.conf
```

Укажите виртуальные хосты в /etc/httpd/conf/extra/httpd-vhosts.conf. Файл уже содержит пример полностью рабочих настроек, что поможет вам быстро выполнить настройки под ваши нужды.

Для проверки виртуальных хостов на локальной машине добавьте их виртуальные имена в файл /etc/hosts:

```
127.0.0.1 domainname1.dom
127.0.0.1 domainname2.dom
```

Перезапустите httpd.service, чтобы изменения вступили в силу.

16.3.7.1. Управление большим количеством виртуальных хостов

Если Apache используется для обслуживания очень большого количества виртуальных хостов, вам может быть полезна возможность их легко включать и отключать. Для этого рекомендуется создавать собственный файл настроек на каждый хост и хранить все эти файлы в одном каталоге, например /etc/httpd/conf/vhosts.

1) Создайте каталог:
```
# mkdir /etc/httpd/conf/vhosts
```
2) Создайте в нем отдельные конфигурационные файлы:
```
# nano /etc/httpd/conf/vhosts/domainname1.dom
# nano /etc/httpd/conf/vhosts/domainname2.dom
...
```
3) Включите эти файлы в основной файл настроек /etc/httpd/conf/httpd.conf:
```
#Enabled Vhosts:
Include conf/vhosts/domainname1.dom
Include conf/vhosts/domainname2.dom
```

Теперь можно быстро включать/отключать требуемые виртуальные хосты, просто закомментировав или раскомментировав соответствующие директивы Include в основном файле настроек.

Очень простой файл виртуального хоста будет выглядеть следующим образом:

```
/etc/httpd/conf/vhosts/domainname1.dom
<VirtualHost *:80>
    ServerAdmin webmaster@domainname1.dom
    DocumentRoot "/home/user/http/domainname1.dom"
    ServerName domainname1.dom
```

```
    ServerAlias domainname1.dom
    ErrorLog "/var/log/httpd/domainname1.dom-error_log"
    CustomLog "/var/log/httpd/domainname1.dom-access_log" common

    <Directory "/home/user/http/domainname1.dom">
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:443>
    ServerAdmin webmaster@domainname1.dom
    DocumentRoot "/home/user/http/domainname1.dom"
    ServerName domainname1.dom:443
    ServerAlias domainname1.dom:443
    SSLEngine on
    SSLCertificateFile "/etc/httpd/conf/server.crt"
    SSLCertificateKeyFile "/etc/httpd/conf/server.key"
    ErrorLog "/var/log/httpd/domainname1.dom-error_log"
    CustomLog "/var/log/httpd/domainname1.dom-access_log" common

    <Directory "/home/user/http/domainname1.dom">
        Require all granted
    </Directory>
</VirtualHost>
```

16.3.7.2. Расширения

PHP

1) Установите пакеты php и php-apache.
2) Чтобы включить PHP, отредактируйте файл /etc/httpd/conf/httpd.conf. В конце списка LoadModule добавьте:

   ```
   LoadModule php7_module modules/libphp7.so
   AddHandler php7-script php
   ```

   В конце списка Include добавьте:

   ```
   Include conf/extra/php7_module.conf
   ```

3) Перезапустите службу httpd.service средствами systemd.

Чтобы убедиться в том, что PHP настроен корректно, создайте файл test.php в каталоге DocumentRoot (то есть в /srv/http/ или ~/public_html) и поместите в него следующий код:

```
<?php phpinfo(); ?>
```

По адресу http://localhost/test.php или http://localhost/"~пользователь"/test.php вы должны увидеть информационную страницу PHP.

Если PHP-код не исполняется, а на странице браузера вы видите содержимое test.php, проверьте, добавили ли вы `Includes` в строку `Options` для вашего корневого

каталога в /etc/httpd/conf/httpd.conf. Кроме того, убедитесь, что `TypesConfig conf/mime.types` раскомментирован в секции `<IfModule mime_module>`. Также можно попробовать добавить следующую строку в секцию `<IfModule mime_module> файла httpd.conf`:

```
AddHandler application/x-httpd-php .php
```

Использование php-fpm и mod_proxy_fcgi

В отличие от широко распространенной установки с ProxyPass, настройка прокси с SetHandler принимает во внимание другие директивы Apache, например, `DirectoryIndex`. Это гарантирует лучшую совместимость с программами, созданными для libphp7, mod_fastcgi и mod_fcgid. Если, тем не менее, необходимо использовать ProxyPass, попробуйте такую строку:

```
ProxyPassMatch ^/(.*\.php(/.*)?)$ unix:/run/php-fpm/php-fpm.sock|
fcgi://localhost/srv/http/$1
```

1) Установите пакет php-fpm.
2) Включите модули прокси:

```
/etc/httpd/conf/httpd.conf
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
```

Создайте файл /etc/httpd/conf/extra/php-fpm.conf следующего содержания:

```
/etc/httpd/conf/extra/php-fpm.conf
<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php-fpm/php-fpm.sock|
fcgi://localhost/"
</FilesMatch>
```

3) Добавьте его в конец файла /etc/httpd/conf/httpd.conf:

```
Include conf/extra/php-fpm.conf
```

Примечание. До и после символа вертикальной линии не должно быть пробелов. Localhost можно заменить любой строкой. Подробности см. в документе по ссылке: https://httpd.apache.org/docs/2.4/mod/mod_proxy_fcgi.html

Можно настроить PHP-FPM в файле /etc/php/php-fpm.d/www.conf, но и параметры по умолчанию должны работать отлично.

Примечание. Если ранее были добавлены следующие строки в httpd.conf, удалите их, т. к. они более не нужны:

```
LoadModule php7_module modules/libphp7.so
Include conf/extra/php7_module.conf
```

4) Перезапустите службы httpd.service и php-fpm.service.

16.3.7.3. Решение проблем

Просмотр журнала и текущего состояния Apache

Чтобы узнать текущее состояние службы httpd, выполните следующую команду:

```
systemctl status httpd
```

Файлы журнала Apache расположены в каталоге /var/log/httpd.

Установленная документация по Apache:

- httpd(8) — руководство по службе httpd с полным списком консольных параметров;
- genkey(1) — руководство для утилиты genkey, поставляемой в пакете crypto-utils;
- apachectl(8) — руководство по Apache HTTP Server Control Interface;
- http://localhost/manual/ — официальная документация для HTTP сервера Apache с полным описанием всех директив и доступных модулей. Обратите внимание, что для чтения этой документации необходимо установить пакет httpd-manual и запустить веб-сервер.

  Перед чтением документации выполните следующие команды:

```
# yum install httpd-manual
# apachectl graceful
```

16.4. Сетевой доступ к файловым системам NFS

Протокол сетевого доступа к файловым системам (NFS) позволяет монтировать файловые системы по сети и взаимодействовать с ними так, как если бы они были смонтированы локально. Это дает возможность системным администраторам консолидировать ресурсы вокруг централизованных серверов в сети.

16.4.1. Введение

На данный момент в ОС РОСА «КОБАЛЬТ» включены две мажорные версии NFS — NFSv3 и NFSv4.0. NFS версии 3 поддерживает защищенный асинхронный режим записи и более устойчива при возникновении ошибок, чем предыдущая NFSv2; также существует поддержка 64-битного размера файлов и смещения, что дает клиентам возможность доступа к файловым данным размером более 2 ГБ. NFS версии 4 работает с межсетевыми экранами и через интернет, не требует службы rpcbind, имеет поддержку ACL и использует операции с сохранением состояния.

В последней версии ОС РОСА «КОБАЛЬТ» добавлена поддержка для NFS версии 4.1 (NFSv4.1). Было произведено несколько улучшений производительности и безопасности, включая клиентскую поддержку для Parallel NFS (pNFS). NFSv4.1 больше не требует отдельного соединения TCP для обратных вызовов, что дает возможность серверу NFS делегировать полномочия, даже если к клиенту невозможно подключиться (например, при вмешательстве со стороны NAT или межсетевого экрана), что предотвращает возможность ранее случавшихся ошибок, когда некоторые операции могли вернуть неточный результат вследствие потери ответа и повторной отправки операции.

По умолчанию клиенты NFS пытаются выполнить монтирование с использованием NFSv4.0, и, если операция оканчивается неудачей, выполняется откат к NFSv3.

Примечание. NFS версии 2 (NFSv2) не поддерживается.

Все версии NFS могут использовать протокол TCP, работающий по сети IP, а для версии NFSv4 он входит в требования. Для обеспечения сетевого соединения без сохранения состояния между клиентом и сервером NFSv3 может использовать протокол UDP, работающий по сети IP.

При использовании NFSv3 с протоколом UDP соединение UDP без сохранения со-

стояния (в нормальных условиях) создает меньше служебной информации, связанной с работой протоколов. Как результат, в очень чистой, неперегруженной сети будет наблюдаться улучшение производительности. Тем не менее, поскольку протокол UDP работает без сохранения состояния, то в случае неожиданного отключения сервера клиенты UDP продолжают наполнять сеть запросами для сервера. Кроме того, при потере кадров в соединении UDP необходимо повторно передать весь запрос RPC целиком, тогда как в TCP нужно переслать только потерянный кадр. По этим причинам при подключении к серверу NFS предпочтительным является протокол TCP.

В протокол NFSv4 были встроены проколы блокировки и монтирования. Сервер также слушает на хорошо известном порту TCP 2049, поэтому для NFSv4 исчезла необходимость взаимодействия с демонами rpcbind, lockd и rpc.statd. Для настройки экспортов на сервере NFS все еще требуется работа демона rpc.mountd, но он не участвует ни в каких операциях передачи данных.

Примечание. В ОС РОСА «КОБАЛЬТ» протоколом по умолчанию для NFSv3 является TCP. Из соображений совместимости можно использовать UDP, но для широкого применения он не рекомендуется. NFSv4 требует TCP.

У всех демонов RPC/NFS существует консольный параметр `-p`, с помощью которого можно указать порт, что облегчает настройку межсетевого экрана. После того, как надстройки TCP получат доступ к клиенту, сервер NFS обращается к файлу /etc/exports, чтобы узнать, разрешен ли клиенту доступ к каким-либо экспортированным файловым системам. После проверки все действия с файлами и каталогами становятся доступными для пользователя.

16.4.2. Требуемые службы

Для предоставления обмена файлами с помощью NFS в ОС РОСА «КОБАЛЬТ» используется сочетание поддержки на уровне ядра и процессов демонов. Все версии NFS зависят от удаленных вызовов процедур (Remote Procedure Calls, RPC) между клиентом и сервером. Службы RPC в ОС РОСА «КОБАЛЬТ» контролируются службой rpcbind. Чтобы смонтировать файловую систему NFS или сделать ее общей, необходима совместная работа следующих служб (в зависимости от реализованной версии NFS):

nfs

Команда `systemctl start nfs` запускает сервер NFS и соответствующие процессы RPC для обслуживания запросов к общим файловым система NFS.

nfslock

Команда `systemctl start nfs-lock` активирует обязательную службу, которая запускает соответствующие процессы RPC, давая возможность клиентам NFS блокировать файлы на сервере.

rpcbind

Служба rpcbind принимает резервирование порта от локальных служб RPC. Затем эти порты анонсируются, чтобы соответствующие удаленные службы RPC поучили к ним доступ. rpcbind отвечает на запросы к службам RPC и настраивает соединения к запрошенным службам RPC. В NFSv4 это не используется.

Следующие процессы RPC облегчают работу служб NFS:

rpc.mountd

Этот процесс используется сервером NFS для обработки запросов MOUNT от клиентов NFSv3. Запрошенный общий ресурс NFS должен быть в текущий момент экспортирован сервером NFS, что и проверяет rpc.mountd, а также проверяется разрешение клиента на доступ к этому общему ресурсу. Если запрос на монтирование разрешается, сервер rpc.mountd посылает в ответ статус Success и отравляет описатель файла этого общего ресурса NFS назад клиенту NFS.

rpc.nfsd

Этот процесс разрешает определение явной версии NFS и протоков, которые анонсирует сервер. Он работает с ядром Linux для удовлетворения потребностей клиентов NFS, таких как предоставление серверных потоков каждый раз при подключении клиента NFS. Этот процесс соответствует службе nfs.

lockd

Это поток ядра, выполняемый как на клиенте, так и на сервере. Он реализует протокол Network Lock Manager (NLM), позволяющий клиентам NFSv3 блокировать файлы на сервере. Он запускается автоматически при каждом запуске сервера NFS и при каждом монтировании файловой системы NFS.

rpc.statd

Этот процесс реализует прокол Network Status Monitor (NSM) RPC, который уведомляет клиентов NFS о перезапуске сервера NFS без корректного его выключения. rpc.statd автоматически запускается службой nfslock, и ему не требуются параметры пользователей. Он не используется с NFSv4.

rpc.rquotad

Этот процесс предоставляет информацию о квоте удаленных пользователей. rpc.rquotad автоматически запускается службой nfs, и ему не требуются параметры пользователей.

rpc.idmapd

rpc.idmapd предоставляет обратные вызовы клиента и сервера NFSv4, которые преобразуются между передаваемыми именами NFSv4 (записи в формате user@domain) и локальными UID и GID. Чтобы idmapd мог функционировать с NFSv4, необходимо настроить файл /etc/idmapd.conf. В минимальной конфигурации должен быть указан параметр «Domain», определяющий домен преобразования NFSv4. Если домен преобразования NFSv4 аналогичен доменному имени DNS, то этот параметр можно опустить. Чтобы преобразование ID функционировало корректно, клиент и сервер должны договориться о домене преобразования NFSv4.

Примечание. В ОС РОСА «КОБАЛЬТ» rpc.idmapd используется только сервером NFSv4. Клиент NFSv4 использует nfsidmap — id-преобразователь на базе связки ключей. nfsidmap — это отдельная программа, вызываемая по требованию ядром для выполнения преобразования ID; это не демон. При наличии проблем с nfsidmap клиент откатывается к использованию rpc.idmapd. Подробные сведения о nfsidmap можно найти на странице руководства nfsidmap.

16.4.3. Настройка клиента NFS

Команда `mount` монтирует общие ресурсы NFS на стороне клиента. Команда имеет следующий формат:

```
# mount -t nfs -o <параметры> <сервер>:</удаленный/экспорт> </локальный/каталог>
```

Здесь:

- `<параметры>` — список параметров монтирования через запятую; подробности о параметрах монтирования NFS см. в п. «Часто используемые параметры монтирования NFS» на стр. 238;
- `<сервер>` — имя хоста, IP-адрес или полное доменное имя сервера, экспортирующего файловую систему, которую нужно смонтировать;
- `</удаленный/экспорт>` — файловая система или каталог, экспортируемый с сервера, то есть каталог, который нужно смонтировать;
- `</локальный/каталог>` — местоположение на клиенте, где смонтирован /remote/export.

Версия протокола NFS, используемого в ОС РОСА «КОБАЛЬТ», определяется параметрами монтирования nfsvers или vers. По умолчанию mount будет использовать NFSv4 в виде `mount -t nfs`. Если сервер не поддерживает NFSv4, клиент автоматически перейдет на версию, используемую сервером. Если параметр nfsvers/vers используется для передачи конкретной версии, не поддерживаемой сервером, монтирование окончится неудачей. Из соображений совместимости с устаревшими версиями также поддерживается тип файловой системы nfs4; это аналогично выполнению команды `mount -t nfs -o nfsvers=4 host:</удаленный/экспорт> </локальный/каталог>`.

Подробности см. на странице руководства mount.

Если общий ресурс NFS был смонтирован вручную, после перезагрузки системы он не будет снова смонтирован автоматически. ОС РОСА «КОБАЛЬТ» предлагает два способа монтирования удаленных файловых систем во время загрузки системы: файл /etc/fstab и служба autofs. Подробности см. в подразделах «Монтирование файловых систем NFS с помощью /etc/fstab и autofs».

16.4.3.1. Монтирование файловых систем NFS с помощью /etc/fstab

Одним из способов монтирования общего ресурса NFS с другой машины является добавление записи в файл /etc/fstab. В записи должны указываться имя хоста сервера NFS, экспортируемый каталог сервера и каталог на локальной машине, куда должен монтироваться общий ресурс NFS. Для изменений файла /etc/fstab требуются права суперпользователя root.

Пример синтаксиса

Общий синтаксис, используемый в строке файла /etc/fstab, выглядит следующим образом:

```
server:/usr/local/pub    /pub    nfs    defaults 0 0
```

Перед выполнением данной команды на клиентской машине должна быть создана

точка монтирования /pub. После добавления указанной строки в файл /etc/fstab на клиентской системе используйте команду mount /pub, и точка монтирования /pub будет смонтирована с сервера.

Действительная запись /etc/fstab для монтирования экспорта NFS должна содержать следующую информацию:

```
<сервер>:/remote/export /local/directory nfs <параметры> 0 0
```

Переменные `<сервер>`, `</удаленный/экспорт>`, `</локальный/каталог>` и `<параметры>` — это те же переменные, которые использовались при ручном монтировании общего ресурса NFS. Определение каждой переменной см. в подразделе «Настройка клиента NFS».

Примечание. Точка монтирования /local/directory должна быть создана на клиенте до чтения файла /etc/fstab. В противном случае монтирование окончится неудачей. Подробности о файле /etc/fstab см. на странице руководства fstab.

16.4.3.2. AUTOFS

Одним из минусов использования /etc/fstab является то, что вне зависимости от того, как часто пользователь получает доступ к смонтированной файловой системе NFS, ОС должна выделять ресурсы для удержания смонтированной файловой системы на месте. Это не является проблемой с одним или двумя смонтированными ресурсами, но когда система должна поддерживать смонтированными множество ресурсов одновременно, то это может отрицательно повлиять на общую производительность. Альтернативой использованию файла /etc/fstab является использование утилиты automount на базе ядра. Средство автоматического монтирования состоит из двух компонентов:

1) Модуля ядра, реализующего файловую систему.
2) Демона в пространстве пользователя, исполняющего другие функции.

Утилита automount может монтировать и размонтировать файловые системы NFS автоматически (монтирование по запросу), тем самым сберегая ресурсы ОС. С ее помощью также можно монтировать другие файловые системы, включая AFS, SMBFS, CIFS и локальные файловые системы.

Примечание. Перед попыткой автоматического монтирования общего ресурса NFS убедитесь в том, что в системе установлены nfs-utils.

Настройка autofs

Основной конфигурационный файл автомонтировщика — /etc/auto.master, также называемый «основной картой». В основной карте перечислены точки монтирования в системе, контролируемые autofs, а также соответствующие конфигурационные файлы или сетевые источники, известные как карты автомонтирования. Основная карта имеет следующий формат:

```
<точка_монтирования> <имя_карты> <параметры>
```

Здесь:

- `<точка_монтирования>` — точка монтирования autofs, например, /home. Это может быть имя отдельного каталога (для непрямого монтирования) или полный путь до точки монтирования (для прямого монтирования). Каждая запись прямого или

непрямого монтирования может быть дополнена списком подкаталогов (каждое имя подкаталога начинается с /, элементы списка разделяются пробелами), что в итоге формирует запись, называемую записью множественного монтирования (multi-mount entry);

- <имя_карты> — имя источника карты, содержащего список точек монтирования и местоположение файловой системы, источника этих точек монтирования. Синтаксис записи карты описывается ниже;
- <параметры> — при их наличии они применяются ко всем записям указанной карты, если у записей отсутствуют собственные указанные параметры. Это поведение отличается от поведения autofs версии 4, где параметры имели накопительный характер. Это поведение было изменено в версии 5, для реализации совместимости со смешанным окружением.

Пример: файл /etc/auto.master

Ниже приведен пример записи из файла /etc/auto.master:

```
/home /etc/auto.misc
```

Общий формат карт аналогичен формату основной карты, но «параметры» размещаются между точкой монтирования и местоположением, в не в конце записи, как в основной карте:

```
<точка_монтирования> [<параметры>] <местоположение>
```

Под <местоположением> имеется в виду местоположение файловой системы, например, путь в локальной файловой системе (перед которым указывается экранирующий символ форматирования карт Sun «:» для имен карт, начинающихся с «/»), в файловой системе NFS или в другом действительном местоположении файловой системы.

Ниже приведен пример содержимого файла карты (например, /etc/auto.misc):

```
payroll -fstype=nfs personnel:/dev/hda3
sales -fstype=ext3 :/dev/hda4
```

Первый столбец в файле карты указывает точку монтирования autofs (sales и payroll с сервера personnel). Второй столбец указывает параметры монтирования autofs, а третий столбец — источник монтирования. Следуя указанной конфигурации, точки монтирования autofs будут: /home/payroll и /home/sales. Параметр `-fstype=` часто опускается и, как правило, не нужен для корректного выполнения операции.

Автомонтировщик создаст два каталога, если они не существуют. Если до запуска автомонтировщика эти каталоги существовали, во время завершения работы автомонтировщик не станет их удалять. Запустить или остановить демон автоматического монтирования можно с помощью одной из следующих команд:

- `service autofs start` (если демон был остановлен);
- `service autofs restart`.

Если процессу нужен доступ к каталогу, который был отмонтирован утилитой autofs, например, /home/payroll/2006/July.sxc, то с помощью вышеуказанной конфигурации демон автоматического монтирования автоматически смонтирует этот каталог. При указанном времени истечения срока ожидания каталог будет автоматически отмонтирован, если за указанный период к нему не был осуществлен доступ.

Статус демона автоматического монтирования можно просмотреть, введя следующую команду:

```
# service autofs status
```

Хранение карт автомонтирования с использованием LDAP

Для возможности получения карт автомонтировщика из LDAP в системе должны быть установлены клиентские библиотеки LDAP. В ОС РОСА «КОБАЛЬТ» пакет openldap должен устанавливаться автоматически как зависимость для автомонтировщика. Для настройки доступа к LDAP измените файл /etc/openldap/ldap.conf. Убедитесь, что BASE, URI и схема имеют параметры, соответствующие конкретному узлу.

Актуальная схема хранения карт автомонтирования в LDAP описана в документе rfc2307bis. Для использования этой схемы ее нужно настроить в параметрах autofs (/etc/sysconfig/autofs), удалив символы комментариев со строк определения схемы.

Пример: настройка параметров autofs

```
DEFAULT_MAP_OBJECT_CLASS="automountMap"
DEFAULT_ENTRY_OBJECT_CLASS="automount"
DEFAULT_MAP_ATTRIBUTE="automountMapName"
DEFAULT_ENTRY_ATTRIBUTE="automountKey"
DEFAULT_VALUE_ATTRIBUTE="automountInformation"
```

Убедитесь в том, что были раскомментированы только строки, соответствующие данной схеме. В схеме rfc2307bis атрибут `automountKey` заменяет атрибут `cn`. LDIF примерной конфигурации приведен ниже.

Пример: конфигурация LDF

```
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (&(objectclass=automountMap)
(automountMapName=auto.master))
# requesting: ALL
#

# auto.master, test.dom
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: top
objectClass: automountMap
automountMapName: auto.master

# extended LDIF
#
# LDAPv3
# base <automountMapName=auto.master,dc=example,dc=com> with
scope subtree
# filter: (objectclass=automount)
```

```
# requesting: ALL
#

# /home, auto.master, test.dom
dn: automountMapName=auto.master,dc=example,dc=com
objectClass: automount
cn: /home

automountKey: /home
automountInformation: auto.home

# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (&(objectclass=automountMap)
(automountMapName=auto.home))
# requesting: ALL
#

# auto.home, test.dom
dn: automountMapName=auto.home,dc=example,dc=com
objectClass: automountMap
automountMapName: auto.home

# extended LDIF
#
# LDAPv3
# base <automountMapName=auto.home,dc=example,dc=com> with scope
subtree
# filter: (objectclass=automount)
# requesting: ALL
#

# foo, auto.home, test.dom
dn: automountKey=foo,automountMapName=auto.home,dc=example,dc=com
objectClass: automount
automountKey: foo
automountInformation: filer.test.dom:/export/foo

# /, auto.home, test.dom
dn: automountKey=/,automountMapName=auto.home,dc=example,dc=com
objectClass: automount
```

```
automountKey: /
automountInformation: filer.test.dom:/export/&
```

Часто используемые параметры монтирования NFS

Кроме монтирования файловой системы на удаленном хосте с помощью NFS, также можно указать другие параметры монтирования для облегчения работы со смонтированным общим ресурсом. Эти параметры можно использовать во время ручного монтирования с помощью команды mount, в конфигурации /etc/fstab, а также с autofs.

Ниже приводятся часто используемые параметры для монтирования ресурсов NFS:

intr

Позволяет прерывать запросы NFS при отключении или недоступности сервера.

lookupcache=<режим>

Указывает ядру, как нужно обрабатывать кэш каталогов для указанной точки монтирования. Действительные аргументы для <режима>: all, none или pos/positive.

nfsvers=<версия>

Указывает, какую версию протокола NFS нужно использовать, где `<версия>` — 3 или 4. Удобно для хостов с несколькими серверами NFS. Если версия не указана, NFS использует самую свежую версию, поддерживаемую ядром и командой mount. Параметр **vers** идентичен параметру **nfsvers** и включен в данную версию из соображений совместимости.

noacl

Отключает обработку ACL. Может понадобиться при работе со старыми версиями различных дистрибутивов Linux или Solaris.

nolock

Отключает блокировку файлов. Этот параметр иногда бывает нужен при подключении к старым серверам NFS.

noexec

Запрещает выполнение бинарных файлов на смонтированных файловых системах. Полезно, если в системе монтируются файловые системы, не принадлежащие семье Linux, содержащие несовместимые бинарные файлы.

nosuid

Отключает биты setuid или sgid, предотвращая получение повышенных привилегий удаленными пользователями с помощью запуска программы setuid.

port=<номер>

Указывает числовое значение порта сервера NFS. Если <номер> равен 0 (значение по молчанию), программа mount запрашивает службу rpcbind удаленного хоста, какой номер порта использовать. Если демон NFS удаленного хоста не зарегистрирован соответствующей службой rpcbind, используется стандартный номер порта TCP — 2049.

rsize= <число> и **wsize= <число>**

Эти параметры ускоряют соединение NFS для чтения (rsize) и записи (wsize), указывая увеличенный размер для передаваемого за один раз блока (<число> в байтах). Изменяйте эти значения осторожно: некоторые старые ядра Linux и сетевые карты не очень

хорошо справляются с увеличенными размерами блоков. Для NFSv3 значения по умолчанию для обоих параметров составляют 8192. Для NFSv4 значения по умолчанию для обоих параметров составляют 32 768.

sec=<режим>

Значение по умолчанию — `sec=sys`, использующее локальные UNIX UID и GID, которые, в свою очередь, используют AUTH_SYS для аутентификации операций NFS.

- `sec=krb5`. Для аутентификации пользователей используется Kerberos V5 вместо локальных UNIX UID и GID;
- `sec=krb5i`. Для аутентификации пользователей используется Kerberos V5, и для предотвращения преднамеренной порчи данных выполняется проверка целостности операций NFS с использованием защищенных контрольных сумм;
- `sec=krb5p`. Для аутентификации пользователей используется Kerberos V5, выполняется проверка целостности, а трафик NFS шифруется для предотвращения прослушивания. Это наиболее надежные параметры, но также и наиболее требовательные к производительности системы.

tcp

Требует использовать протокол TCP для операций монтирования NFS.

udp

Требует использовать протокол UDP для операций монтирования NFS.

Полный список параметров и более подробную информацию о каждом из них см. на страницах руководств mount и nfs.

16.4.4. Запуск и остановка сервера NFS

Для работы сервера NFS, использующего не только версию NFSv4, необходима запущенная служба rpcbind. Для проверки статуса службы rpcbind выполните следующую команду:

```
# systemctl status rpcbind
```

Если служба rpcbind выполняется, службу nfs можно запустить. Чтобы запустить сервер NFS, выполните:

```
# systemctl start nfs
```

Чтобы NFS автоматически запускалась при загрузке системы, выполните:

systemctl enable nfs-server

Примечание. Если служба NFS настроена на запуск при загрузке системы, для версии NFSv3 необходимо также включить службу nfs-lock. В ОС РОСА «КОБАЛЬТ» nfs-lock при необходимости стартует автоматически, и попытка включить ее вручную окончится неудачей. Если же данная служба была отключена, для автоматического запуска nfs-lock при загрузке системы выполните:

```
systemctl enable nfs-lock
```

Чтобы остановить сервер, выполните:

```
# systemctl stop nfs
```

Параметр restart является наиболее быстрым способом остановки и затем переза-

пуска NFS. Это наиболее эффективный способ применить новые параметры после редактирования конфигурационного файла NFS. Чтобы перезапустить сервер, выполните:

```
# systemctl restart nfs
```

После редактирования файла /etc/sysconfig/nfs перезапустите службу nfs-config для применения новых параметров:

```
# systemctl restart nfs-config
```

Команда `try-restart` запускает nfs, только если она уже выполняется. Эта команда является эквивалентом `condrestart` (conditional restart) в сценариях инициализации ОС РОСА «КОБАЛЬТ» и удобна тем, что не запускает демон, если NFS уже выполняется.

Чтобы выполнить условный перезапуск сервера, выполните:

```
# systemctl try-restart nfs
```

Чтобы перезагрузить конфигурацию сервера NFS без перезапуска службы, выполните:

```
# systemctl reload nfs
```

16.4.5. Настройка сервера NFS

Существует два способа настройки экспортов на сервере NFS:

1) Ручное редактирование конфигурационного файла NFS /etc/exports.
2) Использование консольной команды exportfs.

16.4.5.1. Конфигурационный файл /etc/exports

В файле /etc/exports указывается, какие файловые системы экспортируются на удаленный хост. Применяются следующие правила синтаксиса:

- пустые строки игнорируются;
- комментарии начинаются с символа «#»;
- длинные строки переносятся с помощью косой черты «\»;
- для каждой экспортируемой файловой системы выделяется отдельная строка;
- любые списки авторизованных хостов, помещенные после экспортируемой файловой системы, должны отделяться символами пробела;
- параметры для каждого из хостов должны размещаться в скобках и идти непосредственно сразу за идентификатором хоста, без пробела между хостом и первой скобкой.

Каждая запись для экспортируемой файловой системы имеет следующую структуру:

```
export <хост>(<параметры>)
```

В вышеуказанной структуре используются следующие переменные:

- `export` — экспортируемый каталог;
- <хост> — хост или сеть, для которой этот ресурс делается общим;
- `<параметры>` — параметры хоста.

Можно указать несколько хостов, а также параметры для каждого из них. Для этого укажите их в одной строке через пробелы в соответствии со следующим примером:

```
export <хост_1>(<параметры_1>) <хост_2>(<параметры_2>) ...
```

Сведения о разных способах указания имен хостов см. в подразделе «Форматы имен хостов».

В самом простом варианте в файле /etc/exports указываются только экспортируемый каталог и хосты, которым разрешен доступ к этому каталогу. См. пример ниже.

Пример: файл /etc/exports

```
</каталог/экспорта> bob.test.dom
```

Здесь `bob.test.dom` может монтировать `</каталог/экспорта>` с сервера NFS. Поскольку в этом примере никаких параметров не указано, NFS будет использовать параметры по умолчанию.

Параметры по умолчанию:

ro

Экспортируемая файловая система доступна только для чтения. Удаленные хосты не могут изменять общие данные. Чтобы разрешить хостам вносить изменения в файловую систему (то есть читать и писать), укажите параметр rw.

sync

Сервер NFS не будет отвечать на запросы до того, как изменения, сделанные предыдущими запросами, не будут записаны на диск. Чтобы вместо этого включить асинхронную запись, укажите параметр **async**.

wdelay

Сервер NFS отложит запись на диск, если становится очевидным, что скоро поступит еще один запрос записи на диск. Такое поведение может улучшить производительность, т. к. уменьшает количество доступов к диску в результате разрозненных команд записи на диск, тем самым снижая потребление ресурсов записи. Чтобы отключить это поведение, укажите параметр **no_wdelay**. no_wdelay доступен, только если указан параметр по умолчанию sync.

root_squash

Не дает пользователям root, подключенным удаленно, получать привилегии root (в противовес локальным подключениям); вместо этого сервер NFS присвоит им идентификатор пользователя nfsnobody. Этот параметр «выжимает» (squash) привилегии из суперпользователя root, делая его первичным локальным пользователем и предотвращая возможные неавторизованные записи на удаленном сервере. Чтобы отключить это поведение, укажите параметр **no_root_squash**.

Чтобы «выжать» привилегии из каждого удаленного пользователя (включая root), используйте параметр **all_squash**. Чтобы указать идентификаторы пользователя и группы, которые сервер NFS должен присваивать удаленным пользователям с конкретного хоста, используйте соответственно, параметры anonuid и anongid, например:

```
export host(anonuid=uid,anongid=gid)
```

Здесь uid и gid — номер идентификатора пользователя и номер идентификатора группы, соответственно. Параметры `anonuid` и `anongid` дают возможность создать спе-

циальные учетные записи пользователя и группы, для общего использования их удаленными пользователями NFS.

По умолчанию в ОС РОСА «КОБАЛЬТ» служба NFS поддерживает списки контроля доступа (ACL). Для отключения этой возможности при экспорте файловой системы укажите параметр `no_acl`.

Каждое значение по умолчанию для каждой экспортируемой файловой системы должно переписываться явным образом. Если, например, не указывается параметр `rw`, экспортируемая файловая система становится доступной только для чтения. Ниже приведен пример строки из файла /etc/exports, переписывающей два значения по молчанию:

```
</каталог/экспорта> 192.168.0.3(rw,async)
```

В этом примере 192.168.0.3 может монтировать `</каталог/экспорта>` на чтение/запись, и все записи на диск выполняются асинхронно. Подробности о параметрах экспорта см. на странице руководства exportfs.

Другие параметры доступны, если отсутствуют указанные значения по умолчанию. Это относится к возможности отключения проверки поддерева, разрешения доступа с незащищенных портов, а также разрешения незащищенных блокировок файлов (необходимые для некоторых ранних реализаций клиента NFS). Подробности об этих редко используемых параметрах см. на странице руководства exports.

Примечание. Формат файла /etc/exports является очень строгим, особенно относительно использования символа пробела. Не забывайте всегда отделять экспортируемые файловые системы от хостов, а хосты — друг от друга с помощью символа пробела. Но других символов пробела в этом файле быть не должно, за исключением пробелов в комментариях.

Две следующие строки, например, имеют различное значение:

```
/home bob.test.dom(rw)
/home bob.test.dom (rw)
```

Первая строка разрешает доступ на чтение/запись в каталог /home только пользователям с bob.test.dom. Вторая строка разрешает пользователям с bob.test.dom монтировать каталог только для чтения (значение по умолчанию), а все остальные могут монтировать его для чтения-записи.

16.4.6. Команда exportfs

Каждая файловая система, экспортируемая удаленным пользователям с помощью NFS, а также уровень доступа к этим файловым системам, указываются в файле /etc/exports. При запуске службы nfs команда /usr/sbin/exportfs читает этот файл, передает управление фактическим процессом монтирования демону rpc.mountd (если используется версия NFSv3), а затем rpc.nfsd, после чего файловая система становится доступной для удаленных пользователей.

При запуске вручную команда /usr/sbin/exportfs дает пользователю root возможность выборочно экспортировать или отменять экспорт без перезапуска службы NFS. Если указаны корректные параметры, команда /usr/sbin/exportfs записывает экспортируемые файловые системы в /var/lib/nfs/xtab. Поскольку при выдаче прав доступа к файловой си-

стеме rpc.mountd обращается к файлу xtab, изменения в списке экспортируемых файловых систем применяются сразу же.

Ниже перечислены часто используемые параметры команды /usr/sbin/exportfs:

- `-r` — экспортирует все каталоги, перечисленные в /etc/exports, с помощью нового списка экспорта, создаваемого в /etc/lib/nfs/xtab. Этот параметр эффективно актуализирует список экспорта относительно любых изменений, вносимых в /etc/exports;
- `-a` — экспортирует или отменяет экспорт всех каталогов в зависимости от других параметров, переданных команде /usr/sbin/exportfs. При отсутствии других параметров /usr/sbin/exportfs экспортирует все файловые системы, указанные в /etc/exports;
- `-o <файловые_системы>` — указывает каталоги для экспорта, отсутствующие в /etc/exports. Замените `<файловые_системы>` на дополнительные экспортируемые ФС. Формат указания ФС должен совпадать с форматом, используемым в /etc/exports. Этот параметр часто используется для тестирования экспортируемой ФС перед постоянным добавлением ее в список экспортируемых ФС. Подробнее о синтаксисе /etc/exports см. в разделе «Конфигурационный файл /etc/exports»;
- `-i` — файл /etc/exports игнорируется; для определения экспортируемых ФС используются только параметры командной строки;
- `-u` — отменяет экспорт всех общих каталогов. Команда `/usr/sbin/exportfs -ua` приостанавливает процесс доступа к общим файлам NFS, не прерывая работы всех демонов NFS. Чтобы возобновить процесс доступа к общим ресурсам NFS, используйте `exportfs -r`;
- `-v` — подробный отчет о действиях. При выполнении команды `exportfs` будут выводиться гораздо более подробные сведения об экспортируемых файловых системах.

При запуске `exportfs` без параметров она выдает список всех текущих экспортированных файловых систем. Подробнее о команде `exportfs` см. на ее странице руководства.

16.4.6.1. Использование exportfs с NFSv4

В ОС РОСА «КОБАЛЬТ» не требуется специальных шагов для настройки NFSv4, т. к. любые упоминаемые файловые системы автоматически доступны клиентам NFSv3 и клиентам NFSv4 по одному и тому же пути. В предыдущих версиях NFS это было не так. Чтобы запретить клиентам использовать NFSv4, отключите ее, указав в файле /etc/sysconfig/nfs параметр `RPCNFSDARGS= -N 4`.

16.4.7. Работа NFS с межсетевым экраном

Для NFS необходим rpcbind, динамически присваивающий порты службам RPC, что может привести к проблемам при настройке правил межсетевого экрана. Чтобы разрешить клиентам доступ к общим ресурсам NFS за межсетевым экраном, отредактируйте файл /etc/sysconfig/nfs, указав, на каких портах выполняются службы RPC.

Не во всех системах файл /etc/sysconfig/nfs существует по молчанию. Если он отсутствует, создайте его, указав следующее содержимое:

```
RPCMOUNTDOPTS="-p port"
```

Эта запись добавляет `-p port` в командную строку `rpc.mount`.

Чтобы указать, какие порты будут использоваться службой nlockmgr, укажите номер порта для параметров `nlm_tcpport` и `nlm_udpport` в файле /etc/modprobe.d/lockd.conf.

Если NFS сбоит при запуске, проверьте /var/log/messages. Обычно запуск NFS заканчивается неудачей, если был указан уже используемый номер порта. После редактирования /etc/sysconfig/nfs необходимо перезапустить службу nfs-config для применения новых значений:

```
# systemctl restart nfs-config
```

Затем перезапустите сервер NFS:

```
# systemctl restart nfs-server
```

Чтобы проверить, что новые параметры вступили в силу, выполните `rpcinfo -p`.

Примечание. Чтобы разрешить обратным вызовам NFSv4 проход через межсетевой экран, настройте /proc/sys/fs/nfs/nfs_callback_tcpport и разрешите серверу подключаться к этому порту на клиенте.

Для NFSv4 и более поздней это действие не требуется. В окружении, где используется только NFSv4, также не нужны другие порты для mountd, statd и lockd.

16.4.8. Обнаружение экспортируемых каталогов NFS

Для обнаружения файловых систем, экспортируемых NFS, существуют два способа:

1) На любом сервере с поддержкой NFSv3 запустите команду `showmount`:

```
$ showmount -e myserver
Export list for mysever
/exports/foo
/exports/bar
```

2) На любом сервере с поддержкой NFSv4 смонтируйте и просмотрите /:

```
# mount myserver:/ /mnt/
#cd /mnt/
exports
# ls exports
foo
bar
```

На серверах с поддержкой NFSv3 и NFSv4 оба способа приведут к одинаковому результату.

16.4.9. Обеспечение безопасности NFS

NFS является прозрачным механизмом совместного использования целых файловых систем с большим числом известных хостов. Тем не менее, вместе с простотой использования приходит и некоторое число потенциальных проблем безопасности. Чтобы минимизировать риски безопасности при использовании NFS и обеспечить защиту данных на сервере, ознакомьтесь с данным подразделом перед тем, как приступать к экспор-

ту файловых систем NFS или к монтированию их на клиенте.

16.4.9.1. Безопасность NFS с AUTH_SYS и контроль экспорта

Традиционно при использовании NFS существовало две возможности контроля доступа к экспортируемым файлам.

Во-первых, со стороны сервера существуют ограничения касательно того, каким хостам разрешено монтировать какие файловые системы. Хосты при этом идентифицируются по IP-адресу или по имени. Во-вторых, сервер принудительно применяет права доступа к файловым системам для пользователей на клиентах NFS (так же, как это делается и для локальных пользователей). Традиционно это делается с использованием AUTH_SYS (также называемой AUTH_UNIX), которая определяет UID и GID пользователя, основываясь на данных клиента. Необходимо понимать, что в такой ситуации намеренное злоумышленное изменение данных на клиенте может легко привести к нежелательным ситуациям и предоставить пользователю доступ к данным, не предназначенным для него.

Для снижения риска администратор часто ограничивает доступ правами только на чтение или понижает права пользователя до непривилегированного ID пользователя и группы. К сожалению, такие решения не дают использовать NFS так, как исходно задумывалось при ее создании.

Кроме того, если злоумышленник получает контроль над сервером DNS, который используется системой, выполняющей экспорт NFS, систему, связанную с конкретным именем хоста или полным именем, можно направить на неавторизованную машину. На этом этапе неавторизованная машина становится системой, которой разрешено монтировать общий ресурс NFS, поскольку для дополнительной безопасности смонтированного ресурса NFS не требуется обмена информацией о пароле или имени пользователя.

С осторожностью используйте символы подстановки при экспорте каталогов, т. к. в область действия символа подстановки может попасть больше систем, чем планировалось.

Также можно ограничить доступ к службе rpcbind с помощью надстроек TCP. Создание правил iptables также может ограничить доступ к портам, используемым rpcbind, rpc.mountd и rpc.nfsd.

Подробности о том, как защитить NFS и TCP, см. в man iptables.

16.4.9.2. Защита NFS с помощью AUTH_GSS

NFSv4 радикально изменила защиту NFS требованием реализации RPCSEC_GSS и механизма GSS-API пятой версии Kerberos. Тем не менее, RPCSEC_GSS и механизм Kerberos доступны для всех версий NFS. В режиме FIPS можно использовать только алгоритмы, одобренные FIPS.

В отличие от AUTH_SYS, при использовании механизма Kerberos RPCSEC_GSS сервер не зависит от клиента в вопросе правильного представления того, какой именно пользователь получает доступ к файлу. Вместо этого для аутентификации пользователя на сервере используется шифрование, что не дает возможности клиенту-злоумышленнику представиться другим пользователем, не имея учетных данных Kerberos этого пользователя. Использование механизма Kerberos RPCSEC_GSS — это наиболее простой способ обеспечить защиту смонтированным ресурсам, т. к. после настройки Kerberos отпадает

необходимость настраивать что-либо дополнительно.

16.4.9.3. Настройка Kerberos

Перед тем, как настраивать сервер NFSv4, совместимый с Kerberos, необходимо установить и настроить центр распределения ключей Kerberos (Key Distribution Centre, KDC). Kerberos — это система сетевой аутентификации, дающая возможность аутентифицироваться клиентам и серверам с помощью симметричного шифрования и доверенной третьей стороны — центра KDC. Для настройки Kerberos мы рекомендуем использовать управление идентификационной информацией (Identity Management, IdM).

Настройка сервера и клиента NFS для использования RPCSEC_GSS

1) На стороне сервера NFS создайте принципал nfs/hostname.domain@REALM.
2) На стороне клиента и на стороне сервера создайте принципал host/hostname.domain@REALM.
3) Добавьте соответствующие ключи в таблицу ключей клиента и сервера.
4) На стороне сервера включите желаемые степени безопасности, используя sec=<параметр>. Чтобы включить все степени безопасности, а также монтирование ресурсов без шифрования, выполните:

   ```
   /export *(sec=sys:krb5:krb5i:krb5p)
   ```

   Действительные степени безопасности, которые используются с `sec=<параметр>`:

   - **sys**: без защиты шифрованием (значение по умолчанию);
   - **krb5**: только аутентификация;
   - **krb5i**: защита целостности;
   - **krb5p**: защита конфиденциальности.
5) На стороне клиента добавьте в параметры монтирования `sec=krb5`, `sec=krb5i` или `sec=krb5p`, в зависимости от конфигурации:

   ```
   # mount -o sec=krb5 server:/export /mnt
   ```

Хотя мы рекомендуем использовать управление службой идентификации и аутентификации, также поддерживаются серверы Kerberos в Active Directory (AD). Больше информации вы найдете в п. 16.4.11. «Настройка аутентификации Kerberos с использованием SSSD и Active Directory».

Подробные сведения см. на страницах руководств exports(5) и nfs(5), а также в п. «Часто используемые параметры монтирования NFS» на стр. 238.

16.4.9.4. Защита NFS в версии NFSv4

В NFSv4 поддержка ACL реализована на базе модели Microsoft Windows NT, а не модели POSIX, в связи с возможностями модели Windows NT и ее широким применением.

Другой важной возможностью безопасности NFSv4 является отказ от использования прокола MOUNT для монтирования файловых систем. Протокол MOUNT представлял возможные уязвимости для безопасности из-за итого, каким образом он обрабатывал дескрипторы файлов.

16.4.9.5. Права доступа к файлам

После того, как файловая система NFS смонтирована удаленным хостом с правами

на чтение-запись, единственной защитой для каждого общего файла являются его права доступа. Если одна и та же файловая система NFS будет смонтирована двумя пользователями с одинаковыми записями идентификатора пользователя, то эти пользователи смогут изменять файлы друг друга. Кроме того, любой пользователь, авторизованный в системе с правами root, может использовать команду su для доступа к любым файлам в пределах общего ресурса NFS.

По умолчанию в ОС РОСА «КОБАЛЬТ» активированы списки контроля доступа (ACL). Мы рекомендуем оставить эту возможность включенной.

По умолчанию при экспорте файловой системы NFS использует понижение привилегий для root, что устанавливает идентификатор пользователя любого, получившего доступ к общему ресурсу NFS с правами root локальной машины, на уровень nobody. Понижение привилегий root контролируется параметром по умолчанию `root_squash`; подробности об этом параметре см. в подразделе «Конфигурационный файл /etc/exports». Рекомендуется никогда не отключать понижение привилегий root.

При экспорте общего ресурса NFS на чтение/запись лучше использовать параметр all_squash. Этот параметр принудительно выдает каждому пользователю, получающему доступ к экспортированной файловой системе, идентификатор пользователя nfsnobody.

16.4.10. NFS и RPCBIND

Примечание. Сведения в данном разделе касаются только реализации NFSv3, для которой требуется служба rpcbind для обратной совместимости.

Утилита rpcbind отображает службы RPC на порты, на которых эти службы слушают. Процессы RPC уведомляют rpcbind о своем запуске, регистрируя порты, на которых они слушают, и сообщая программные номера RPC, которые они собирается обслуживать. Далее клиентская система передает rpcbind на сервере конкретный программный номер RPC. Служба rpcbind перенаправляет клиента на соответствующий номер порта, чтобы состоялся обмен информацией требуемой службой.

Поскольку для создания соединений со входящими клиентскими запросами службам на основе RPC требуется rpcbind, rpcbind должен быть доступен в системе до запуска любой из этих служб.

Для контроля доступа служба rpcbind использует надстройки TCP, а правила контроля доступа для rpcbind влияют на все службы на основе RPC. Как вариант, можно указать правила контроля доступа для каждого из демонов NFS RPC. Точный синтаксис написания этих правил можно найти на страницах руководств rpc.mountd и rpc.statd.

16.4.10.1. Поиск и устранение проблем с NFS и rpcbind

Поскольку rpcbind предоставляет координацию между службами RPC и номерами портов, используемых для обмена с ними информацией, то при решении проблем бывает полезным просмотреть статус текущих служб RPC с помощью rpcbind. Команда rpcinfo показывает информацию о каждой службе на базе RPC с номерами портов, программный номер RPC. Номер версии и тип протокола IP (TCP или UDP).

Чтобы убедиться в том, что для rpcbind включены сочувствующие службы NFS на базе RPC, выполните следующую команду:

```
# rpcinfo -p
```

Пример: вывод команды rpcinfo -p

```
program vers proto port service
      100021    1   udp  32774  nlockmgr
      100021    3   udp  32774  nlockmgr
      100021    4   udp  32774  nlockmgr
      100021    1   tcp  34437  nlockmgr
      100021    3   tcp  34437  nlockmgr
      100021    4   tcp  34437  nlockmgr
      100011    1   udp    819  rquotad
      100011    2   udp    819  rquotad
      100011    1   tcp    822  rquotad
      100011    2   tcp    822  rquotad
      100003    2   udp   2049  nfs
      100003    3   udp   2049  nfs
      100003    2   tcp   2049  nfs
      100003    3   tcp   2049  nfs
      100005    1   udp    836  mountd
      100005    1   tcp    839  mountd
      100005    2   udp    836  mountd
      100005    2   tcp    839  mountd
      100005    3   udp    836  mountd
      100005    3   tcp    839  mountd
```

Если одна из служб не запускается корректно, rpcbind будет не в состоянии отобразить запросы RPC от клиентов к этой службе на правильный порт. Во многих случаях, когда в выводе rpcinfo отсутствует NFS, перезапуск NFS заставляет службу корректно зарегистрироваться в rpcbind и начать работу.

Подробную информацию и список параметров rpcinfo см. на соответствующей странице руководства.

16.4.11. Настройка аутентификации Kerberos с использованием SSSD и Active Directory

16.4.11.1. Принципалы Kerberos и Active Directory

Active Directory разделяет принципалы Kerberos на две категории: принципалы пользователей и принципалы службы. Принципалы пользователей используются в процессе обмена службы аутентификации (AS) для получения мандата на получение мандата (TGT, Ticket Granting Ticket), а принципалы службы получаются в процессе обмена TGS (службы, предоставляющей мандаты, Ticket Granting Service) с использованием TGT. Учетная запись Active Directory может иметь множество принципалов службы, но только один принципал пользователя. Как результат, даже если в таблице ключей есть ключ для принципала, получить TGT с использованием этого принципала (например, через kinit) можно, только если этот принципал присутствует в поле `userPrincipalName` учетной за-

писи компьютера в Active Directory. Помимо этого, можно всегда получить TGT, используя принципал, совпадающий с атрибутом `sAMAccountName` в Active Directory (при условии, что для этого принципала есть ключ в таблице ключей). Обычно `sAMAccountName` — это короткое имя хоста с добавленным в начале символом $.

Также обратите внимание, что если при присоединении к Active Directory с помощью команды `net ads join` был указан параметр createupn, это действие обновляет только поле `userPrincipalName` в Active Directory. Поле `servicePrincipalName` нужно будет обновить либо с помощью инструмента «Пользователи и компьютеры Active Directory» в графическом интерфейсе пользователя, либо с помощью команды setspn.exe в консоли.

16.4.11.2. Принципалы Kerberos и NFS

Клиент NFS выполняет поиск используемого ключа в таблице ключей в таком порядке (см. страницу руководства rpc.gssd):

```
<hostname>$@<REALM>
<HOSTNAME>$@<REALM>
root/<hostname>@<REALM>
nfs/<hostname>@<REALM>
host/<hostname>@<REALM>
root/<anyname>@<REALM>
nfs/<anyname>@<REALM>
host/<anyname>@<REALM>
```

Клиент NFS сделает попытку получить TGT с помощью первого же найденного совпадающего принципала. Если с помощью этого принципала получить TGT невозможно (например, этот принципал не совпадает с атрибутами учетной записи компьютера `sAMAccountName` или `userPrincipalName` в Active Directory), то тогда попытка создания контекста безопасности с сервером NFS будет неудачной. Клиент NFS **не** станет продвигаться по списку к следующему принципалу.

Хотя клиент NFS может использовать различные принципалы, сервер NFS выполнит обратный запрос NFSv4 для клиента только если клиент использовал принципал службы (смотрите последний блок в `gssp_accept_sec_context_upcall` в `net/sunrpc/auth_gss/gss_rpc_upcall.c` кода ядра).

Сервер NFS будет использовать только принципала службы NFS `nfs/<hostname>@<REALM>`.

И наконец, когда сервер NFS выполняет обратный запрос клиента NFS, по сути, сервер становится клиентом, а клиент становится сервером. Это означает, что для использования обратных вызовов в NFS с настроенным Kerberos как на клиенте, так и сервере должны выполняться и rpc.gssd, и gssproxy.

Администратор должен решить, использовать ли делегирование NFSv4. Если в делегировании NFSv4 нет необходимости, не нужно будет беспокоиться о том, сможет ли сервер выполнить обратный запрос к клиенту, и процесс настройки немного упростится.

Для обеспечения максимальной функциональности протокола NFSv4 мы рекомендуем настраивать использование принципалов служб NFS как на клиентах, так и на серверах NFS.

16.4.11.3. Шаги настройки, универсальные для всех машин

1) Установите необходимые пакеты (обратите внимание, что будут также установлены и пакеты с зависимостями):

```
yum install realmd krb5-workstation sssd adcli samba-common
oddjob oddjob-mkhomedir
```

2) Настройте параметры Kerberos в файле etc/krb5.conf:

```
includedir /var/lib/sss/pubconf/krb5.include.d/
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log
[libdefaults]
default_realm = test.dom
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
[realms]
test.dom = {
kdc = addc.test.dom
admin_server = addc.test.dom
}
[domain_realm]
.test.dom = test.dom
test.dom = test.dom
```

3) Получите TGT Kerberos для пользователя, имеющего права на присоединение к компьютерам в домене:

```
# kinit Administrator
```

4) Перед присоединением к домену выполните `realm discover`:

```
# realm discover test.dom
test.dom
type: kerberos
realm-name: test.dom
domain-name: test.dom
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
```

```
required-package: adcli
required-package: samba-common
```

5) Присоединитесь к домену:

   - Если делегирование NFSv4 **не нужно**:

     ```
     # realm join test.dom
     ```

   - Если делегирование NFSv4 **нужно**:

     ```
     # realm join --user-principal=nfs/server.test.dom@test.dom
     test.dom
     ```

Примечания.

1. В вышеуказанной команде используется условное имя хоста сервера; при выполнении на сервере вставьте соответствующее значение.

2. Параметр `-user-principal` обновляет поле `userPrincipalName` учетной записи компьютера в Active Directory. В отличие от поля `servicePrincipalName`, являющегося списком, поле `userPrincipalName` может содержать только один принципал.

3. Команда `realm join` автоматически обновляет параметры SSSD, PAM и nsswitch, а также запускает службу SSSD.

16.4.11.4. Шаги настройки для серверов NFS

6) Выполните вход на контролер домена Active Directory и запустите команду `setspn` для добавления принципала службы nfs сервера:

```
PS C:\Users\Administrator> setspn -A nfs/server.test.dom server
Checking domain DC=example,DC=com
Registering ServicePrincipalNames for
CN=server,CN=Computers,DC=example,DC=com
nfs/server.test.dom
Updated object
PS C:\Users\Administrator> setspn -L server
Registered ServicePrincipalNames for
CN=server,CN=Computers,DC=example,DC=com:
nfs/server.test.dom
HOST/server.test.dom
HOST/SERVER
PS C:\Users\Administrator>
```

7) Запустите демон(ы) GSS.

Примечание. В ОС РОСА «КОБАЛЬТ» обработка данных учетных записей Kerberos со стороны потоков ядра nfsd на сервере NFS, а также работа службы обратных вызовов NFSv4 на клиенте NFS теперь контролируется демоном gssproxy, а не rpc.svcgssd.

- Если делегирование NFSv4 **не нужно**:

  Проверьте, что gssproxy выполняется. По молчанию, она уже должна выполняться, и ее не нужно включать в systemd.

  ```
  # systemctl status gssproxy
  ```

- Если делегирование NFSv4 **нужно**:

Проверьте, что gssproxy выполняется и запустите службу rpc-gssd. Это нужно только непосредственно после присоединения к домену. При последующих перезапусках обе службы ложны запускаться автоматически, и ни одну из них не нужно включать в systemd.

```
# systemctl status gssproxy
# systemctl start rpc-gssd
```

8) Создайте запись в /etc/exports:

```
/export *(rw,sec=krb5:krb5i:krb5p)
```

16.4.11.5. Шаги настройки для клиентов NFS

6) Если делегирование NFSv4 **не нужно**, переходите к шагу 10). В противном случае войдите на контролер домена Active Directory и выполните команду `setspn` для добавления принципала службы nfs для клиента:

```
PS C:\Users\Administrator> setspn -A nfs/client.test.dom client
Checking domain DC=example,DC=com
Registering ServicePrincipalNames for
CN=client,CN=Computers,DC=example,DC=com
nfs/client.test.dom
Updated object
PS C:\Users\Administrator> setspn -L client
Registered ServicePrincipalNames for
CN=client,CN=Computers,DC=example,DC=com:
nfs/client.test.dom
HOST/client.test.dom
HOST/CLIENT
PS C:\Users\Administrator>
```

7) Для создания таблицы ключей, содержащей только принципала nfs, используйте команду `ktutil` (это нужно, чтобы rpc-gssd обязательно первым нашла принципала службы nfs):

Несколько полезных советов по использованию ktutil

- у ktutil есть только несколько простых команд, которые можно просмотреть с помощью «?»;
- при использовании команды delent остальные элементы списка перемещаются вверх по списку. По этой причине лучше начинать с последней записи, которую нужно удалить, и продвигаться дальше вверх;
- при записи файла таблицы ключей с использованием команды wkt, то если файл таблицы ключей уже существует, то записи в памяти будут добавлены в файл. Перед выполнением записи нужно сначала проверить, существует ли таблица ключей, если такое поведение нежелательно.

```
[root@client ~]# ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: l
slot KVNO Principal
```

```
---- ----
------------------------------------------------------------------
----
1
2
host/client.test.dom@test.dom
2
2
host/client.test.dom@test.dom
3
2
host/client.test.dom@test.dom
4
2
host/client.test.dom@test.dom
5
2
host/client.test.dom@test.dom
6
2
host/client@test.dom
7
2
host/client@test.dom
8
2
host/client@test.dom
9
2
host/client@test.dom
10
2
host/client@test.dom
11
2
CLIENT$@test.dom
12
2
CLIENT$@test.dom
13
2
CLIENT$@test.dom
14
```

```
2
CLIENT$@test.dom
15
2
CLIENT$@test.dom
16
2
nfs/client.test.dom@test.dom
17
2
nfs/client.test.dom@test.dom
18
2
nfs/client.test.dom@test.dom
19
2
nfs/client.test.dom@test.dom
20
2
nfs/client.test.dom@test.dom
ktutil: delent 15
ktutil: delent 14
ktutil: delent 13
ktutil: delent 12
ktutil: delent 11
ktutil: delent 10
ktutil: delent 9
ktutil: delent 8
ktutil: delent 7
ktutil: delent 6
ktutil: delent 5
ktutil: delent 4
ktutil: delent 3
ktutil: delent 2
ktutil: delent 1
ktutil: l
slot KVNO Principal
---- ----
------------------------------------------------------------------
----
1
2
nfs/client.test.dom@test.dom
```

```
2
2
nfs/client.test.dom@test.dom
3
2
nfs/client.test.dom@test.dom
4
2
nfs/client.test.dom@test.dom
5
2
nfs/client.test.dom@test.dom
ktutil: wkt /etc/nfs.keytab
ktutil: q
[root@client ~]#
```

8) Если делегирование NFSv4 **нужно**, отредактируйте файл etc/sysconfig/nfs, раскомментировав строку RPCGSSDARGS и добавив следующее:

   ```
   RPCGSSDARGS="-k /etc/nfs.keytab"
   ```

9) Обновите конфигурацию NFS:

   ```
   systemctl restart nfs-config
   ```

10) Запустите демоны GSS.

    - Если делегирование NFSv4 **не нужно**:

      Запустите службу rpc-gssd. Это нужно только непосредственно после присоединения к домену. Во время последующих перезапусков служба должна запускаться автоматически, и включать ее в systemd не нужно.

      ```
      # systemctl start rpc-gssd
      ```

    - Если делегирование NFSv4 **нужно**:

      Проверьте, что gssproxy выполняется, и запустите службу rpc-gssd. Это необходимо только непосредственно после присоединения к домену. Во время последующих перезапусков обе службы должны запускаться автоматически, и включать их в systemd не нужно.

      ```
      # systemctl status gssproxy
      # systemctl start rpc-gssd
      ```

Примечание. В текущей версии ОС РОСА «КОБАЛЬТ» обработка данных учетных записей Kerberos со стороны потоков ядра nfsd на сервере NFS, а также работа службы обратных вызовов NFSv4 на клиенте NFS контролируются демоном gssproxy, а не rpc.svcgssd.

Теперь у системного администратора ОС РОСА «КОБАЛЬТ» есть возможность монтировать файловую систему NFS с помощью `sec=krb5`, `sec=krb5i` или `sec=krb5p`.

Доступная локально документация по NFS

Администрирование сервера NFS может быть нелегкой задачей. Для экспорта или монтирования общих ресурсов NFS существует много параметров, включая в том числе и

упомянутые в данном руководстве. Подробности ищите на страницах следующих руководств:

- man mount — исчерпывающая информация о параметрах mount как для конфигурации сервера, так и для конфигурации клиента NFS;
- man fstab — сведения о формате файла /etc/fstab, который используется при монтировании файловых систем при загрузке системы;
- man nfs — подробности об операциях экспорта и импорта файловых систем NFS;
- man exports — общие параметры, используемые в файле /etc/exports для экспорта файловых систем NFS.

16.5. Samba

Samba — это стандартный набор свободных программ Linux для взаимодействия с ОС Windows, реализующий сетевой протокол Server Message Block (SMB). Протокол SMB предоставляет Microsoft Windows, Linux, Unix и другим ОС возможность доступа к общим файлам и принтерам с помощью серверов, поддерживающих этот протокол. Благодаря протоколу SMB клиенты Windows воспринимают Samba как сервер Windows.

Примечание. Чтобы иметь возможность работать с Samba, убедитесь, что пакет samba установлен в системе. Для этого выполните следующую команду с привилегиями суперпользователя root:

```
# yum install samba
```

16.5.1. Введение

Samba является важным компонентом бесшовной интеграции серверов и рабочих станций Linux в окружение Active Directory (AD). Она может выполнять роль как контроллера домена (в стиле NT4), так и обычного члена домена (в стиле Active Directory или NT4).

Что может Samba:

- обеспечивать доступ к деревьям каталога и принтерам для клиентов Linux, UNIX и Windows;
- помогать в просмотре сети (с NetBIOS);
- выполнять аутентификацию для входа в домен Windows;
- предоставлять разрешение имен сервера доменных имен Windows (WINS);
- выступать первичным контроллером домена (PDC) в стиле Windows NT®;
- выступать резервным контроллером домена (BDC) для первичных контроллеров домена на базе Samba;
- выполняеть роль сервера-члена домена Active Directory;
- присоединяться к Windows NT/2000/2003/2008 PDC/Windows Server 2012.

Чего не может Samba:

- выступать резервным контроллером домена (BDC) для первичного контроллера домена на базе Windows (PDC) (и наоборот);
- выполнять роль контроллера домена Active Directory.

Примечание. Перед установкой и настройкой SMB-сервера необходимо убедиться, что в файле /etc/hosts прописано верное FDQN-имя сервера. Для проверки выполните следующую команду:

```
hostname -f
```

Убедитесь, что в ответ на нее система не выдает ошибку. В противном случае отредактируйте файл /etc/hosts, внеся в него следующую строку:

```
<IP-адрес> <каноническое_имя_сервера> <псевдоним(ы)>
```

16.5.2. Демоны и службы Samba

Samba базируется на работе трех демонов — smbd, nmbd и winbindd. Три службы — smb, nmb и winbind — контролируют процесс запуска и остановки демонов, а также выполняют другие служебные действия. Эти службы играют роль сценариев инициализации. Каждый демон подробно описывается ниже, а также указывается, какая именно служба его контролирует

smbd

Серверный демон smbd предоставляет услуги по доступу к общим файлам и принтерам для клиентов Windows. Кроме того, он отвечает за аутентификацию пользователей, блокирование ресурсов и предоставление доступа к общим данным с использованием протокола SMB. Порты по умолчанию, на которых сервер слушает трафик SMB — порты TCP 139 и 445.

Демон smbd контролируется службой smb.

nmbd

Серверный демон nmbd понимает запросы службы имен NetBIOS и отвечает на них. Это могут быть, например, запросы SMB/CIFS в системах на базе Windows. Также демон принимает участие в протоколах просмотра, задействованных в сетевом окружении Windows. Порт по умолчанию, на котором сервер слушает трафик NMB, это UDP 137.

Демон nmbd контролируется службой nmb.

winbindd

Служба winbind разрешает информацию пользователя и группы, полученную с сервера, на котором установлена ОС Windows NT, 2000, 2003, Windows Server 2008 или Windows Server 2012, и делает эту информацию понятной для платформ UNIX. Это достигается использованием вызовов Microsoft RPC, модулей PAM и NSS. В итоге, пользователи домена Windows NT и Active Directory представляются и действуют, как пользователи UNIX на машине UNIX. Несмотря на то, что служба winbind идет в программном составе Samba, она контролируется отдельно от службы smb.

Демон winbind контролируется службой winbind и не требует для своей работы запуска службы smb. winbind также используется с Samba в качестве члена Active Directory, и также может использоваться на контроллере домена Samba (для реализации вложенных групп и доверия между доменами).

Примечание. Список утилит, включенных в состав Samba, можно посмотреть в подразделе «Программы в составе Samba».

16.5.3. Подключение к общему ресурсу Samba с помощью smbclient

Утилита smbclient дает возможность подключаться к общему ресурсу SMB и выполнять действия, аналогичные действиям клиента FTP. Чтобы, например, подключиться к Demo_Share на хосте SMB-Server и выполнить аутентификацию с использованием имени пользователя administrator, выполните следующую команду:

```
# smbclient //SMB-Server/Demo_Share -Uadministrator
```

После удачного входа введите `help` для просмотра списка доступных команд:

```
smb:\> help
```

Чтобы, например, перейти в каталог Example, выполните:

```
smb:\> cd Example
```

Для отключения выполните:

```
smb:\> exit
```

16.5.4. Монтирование общего ресурса

Иногда бывает удобным смонтировать общий ресурс Samba в каталог, чтобы файлы в каталоге обрабатывались так, как будто они являются частью локальной файловой системы.

Чтобы смонтировать общий ресурс Samba в каталог, создайте каталог для монтирования (если он еще не существует) и выполните следующую команду с привилегиями суперпользователя root:

```
mount -t cifs //servername/sharename /mnt/point/ -o
username=username,password=password
```

Эта команда монтирует общий ресурс sharename с сервера servername в локальный каталог /mnt/point/.

Подробную информацию о монтировании общих ресурсов Samba см. на странице руководства mount.cifs(8).

Примечание. Утилита mount.cifs поставляется в отдельном пакете RPM (независимо от Samba). Перед использованием mount.cifs убедитесь, что в системе установлен пакет cifs-utils. выполните следующую команду с привилегиями суперпользователя root:

```
# yum install cifs-utils
```

Обратите внимание, что в пакет cifs-utils также включен бинарный файл cifs.upcall, вызываемый ядром для монтирования CIFS с настроенной поддержкой Kerberos. Подробности о cifs.upcall см. на странице руководства cifs.upcall(8).

Примечание. Некоторые серверы CIFS для аутентификации требуют незашифрованные пароли в открытом виде. Поддержку простых текстовых паролей можно включить с помощью следующей команды, введенной с привилегиями суперпользователя root:

```
# echo 0x37 > /proc/fs/cifs/SecurityFlags
```

Отмена шифрования паролей может стать причиной утечки паролей!

16.5.5. Настройка сервера Samba

Конфигурационный файл по умолчанию (/etc/samba/smb.conf) дает пользователям

возможность рассматривать свои домашние каталоги как общий ресурс Samba. Также общими ресурсами Samba становятся настроенные в системе принтеры. На подключенный к системе принтер можно посылать задания печати с сетевых машин Windows.

16.5.5.1. Конфигурация в командной строке

В качестве конфигурационного файла Samba использует /etc/samba/smb.conf. Изменения, вносимые в этот файл, не применяются до перезапуска демона Samba:

```
# systemctl restart smb.service
```

Чтобы указать рабочую группу Windows и краткое описание сервера Samba, добавьте следующие записи в файл /etc/samba/smb.conf:

```
workgroup = WORKGROUPNAME
server string = <краткий_комментарий_о_сервере>
```

Замените `WORKGROUPNAME` именем рабочей группы Windows, к которой должна принадлежать данная машина. `<краткий_комментарий_о_сервере>` необязателен и используется в качестве комментария Windows для системы Samba. Чтобы создать общий каталог Samba в системе Linux, добавьте следующий раздел в файл /etc/samba/smb.conf (после того, как в него были внесены изменения, отражающие требования и параметры системы):

Пример: типовая конфигурация сервера Samba

```
[sharename]
comment = Insert a comment here
path = /home/share/
valid users = ivanov petrov
writable = yes
create mask = 0765
```

В примере выше пользователям ivanov и petrov разрешается читать и писать в каталоге /home/share/ на сервере Samba с клиента Samba.

16.5.5.2. Шифрование паролей

Шифрование паролей включено по умолчанию как более надежный вариант защиты. Чтобы создать пользователя с зашифрованным паролем, используйте утилиту smbpasswd:

```
smbpasswd -a username
```

16.5.6. Запуск и остановка Samba

Чтобы запустить сервер Samba, выполните следующую команду с привилегиями суперпользователя root:

```
# systemctl start smb.service
```

Примечание. Чтобы настроить сервер как член домена, перед запуском службы smb нужно присоединиться к домену или к Active Directory с помощью команды net join. Кроме того, перед запуском smbd рекомендуется запустить winbind.

Для остановки сервера выполните:

```
# systemctl stop smb.service
```

Параметр restart — быстрый способ остановить и затем сразу снова запустить Samba. Это самый надежный способ применить изменения, внесенные в конфигурацию Samba. Обратите внимание, что параметр restart запустит демон, даже если до этого он не был запущен.

Чтобы перезапустить сервер, выполните:

```
# systemctl restart smb.service
```

Параметр `condrestart` (conditional restart, условный перезапуск) перезапустит smb, только если он уже выполняется. Этот параметр удобен для сценариев, т. к. демон не будет запущен, если он не выполняется.

Примечание. Если в файл /etc/samba/smb.conf были внесены изменения, Samba автоматически перезагрузит его через несколько минут. Перезапуск или перезагрузка вручную имеет такой же результат.

Для условного перезапуска сервера выполните:

```
# systemctl try-restart smb.service
```

Ручная перезагрузка файла /etc/samba/smb.conf может пригодиться в случае сбоя автоматической перезагрузки со стороны службы smb.

```
# systemctl reload smb.service
```

По умолчанию служба smb не запускается автоматически при загрузке системы. Чтобы настроить автоматический запуск службы, выполните:

```
# systemctl enable smb.service
```

16.5.7. Режимы безопасности Samba

Для Samba существуют два типа защиты — share-level и user-level — которые в совокупности известны как «уровни безопасности». Защита share-level устарела и была удалена из Samba, конфигурации с этим уровнем защиты необходимо обновить на использование user-level. Защита user-level может быть реализована одним из трех различных способов, которые называются режимами безопасности.

16.5.7.1. Защита user-level

Защита уровня user-level применяется по умолчанию и является рекомендуемым уровнем для Samba. Даже если директива `security = user` не указана в файле /etc/samba/smb.conf, она используется в Samba. Если сервер принимает имя пользователя и пароль клиента, то клиент может затем смонтировать несколько общих ресурсов без указания пароля для действий с каждым из них. Samba также принимает запросы имени пользователя и пароля на сеансовой основе. Клиент поддерживает несколько контекстов аутентификации, используя уникальный UID для каждого входа.

В файле /etc/samba/smb.conf директива security = user, настраивающая защиту уровня user-level, имеет следующий вид:

```
[GLOBAL]
...
security = user
...
```

16.5.7.2. Гостевые ресурсы Samba

Как упоминалось выше, режим защиты share-level является устаревшим. Чтобы настроить гостевой ресурс Samba без применения параметра security = share, выполните следующие шаги:

Настройка гостевых ресурсов Samba

1) Создайте файл отображения имени пользователя, в нашем примере это /etc/samba/smbusers, и добавьте в него следующую запись:

```
nobody = guest
```

2) Добавьте следующие директивы в главный раздел файла /etc/samba/smb.conf. Не используйте директив действительных пользователей:

```
[GLOBAL]
...
security = user
map to guest = Bad User
username map = /etc/samba/smbusers
...
```

Директива username map указывает путь к файлу отображения пользователей, упоминаемому в предыдущем шаге.

3) Добавьте следующую директиву в раздел share файла /ect/samba/smb.conf. Не используйте директив действительных пользователей.

```
[SHARE]
...
guest ok = yes
...
```

Ниже описываются другие реализации защиты уровня user-level.

16.5.7.3. Режим domain security (уровень User-Level)

В режиме защиты domain сервер Samba имеет учетную запись компьютера (учетная запись domain security trust) и принудительно направляет все запросы авторизации через контроллеры доменов. Сервер Samba превращается в сервер-член домена с помощью следующих директив в файле /etc/samba/smb.conf:

```
[GLOBAL]
...
security = domain
workgroup = MARKETING
...
```

16.5.7.4. Режим безопасности Active Directory (уровень user-level)

В окружении Active Directory есть возможность присоединения к домену в качестве естественного члена Active Directory. Даже если политика безопасности запрещает использование протоколов аутентификации, совместимых с NT, сервер Samba может присоединиться к Active Directory при помощи Kerberos. Samba в режиме члена Active Directory может принимать билеты Kerberos.

Следующие директивы в файле /etc/samba/smb.conf делают Samba членом сервера Active Directory:

```
[GLOBAL]
...
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
...
```

16.5.7.5. Безопасность уровня share-level

На уровне безопасности share-level сервер принимает от клиента только пароль, без явно указанного имени пользователя. Для каждого ресурса сервер создает пароль, независимо от имени пользователя. Известны случаи, когда клиенты Microsoft Windows сталкивались с проблемами совместимости при работе с серверами с уровнем безопасности share-level. Этот режим устарел и был удален из Samba. Конфигурации, содержащие `security = share`, должны быть обновлены для использования уровня user-level. Чтобы отказаться от использования директивы `security = share`, следуйте шагам, описанным в подразделе «Настройка гостевых ресурсов Samba».

16.5.8. Просмотр сетевых ресурсов Samba

Возможность просмотра сетевых ресурсов позволяет серверам Samba и Windows присутствовать в «Сетевом окружении» Windows. Внутри «Сетевого окружения» значки представляют серверы, и при их открытии показываются доступные общие ресурсы и принтеры сервера.

Возможности просмотра сети требуют реализации NetBIOS по TCP/IP. Для возможности управления списком просмотра ресурсов, в сетях на базе NetBIOS используются широковещательные сообщения (UDP). В отсутствие таких наиболее очевидных способов для разрешения имен хостов TCP/IP, как NetBIOS и WINS, необходимо использовать другие способы, например статичные файлы (/etc/hosts) или DNS.

Главный обозреватель сети домена собирает и сравнивает списки просмотра локальных главных обозревателей всех подсетей, так чтобы просмотр сети можно было осуществлять между всеми рабочими группами и подсетями. Кроме того, главный обозреватель домена предпочтительно должен быть локальным главным обозревателем в своей подсети.

16.5.8.1. Просмотр доменов

По умолчанию Windows-первичный контроллер домена также является главным обозревателем сети этого домена. Сервер Samba не должен настраиваться как главный сервер домена в таких ситуациях.

Для подсетей, в которых отсутствует первичный контроллер домена под управлением Windows, сервер Samba может быть реализован как локальный главный обозреватель. Параметры файла /etc/samba/smb.conf для главного локального обозревателя сети (или совсем без просмотра сети) в окружении контроллера домена совпадают с параметрами для рабочей группы (см. подраздел «Настройка сервера Samba»).

16.5.9. WINS (Windows Internet Name Server)

В качестве сервера WINS может выступать либо сервер Samba, либо сервер Windows NT. Если сервер WINS используется с включенном NetBIOS, одноадресные передачи UDP можно маршрутизировать, что позволяет использовать разрешение имен между различными сетями. В отсутствии сервера WINS передача UDP ограничена локальной подсетью и, соответственно, не может быть маршрутизирована в другие подсети, рабочие группы или домены. Если необходимо использовать репликацию WINS, не используйте Samba в качестве первичного сервера WINS, поскольку на настоящее время Samba не поддерживает репликацию WINS.

В смешанном окружении Samba и сервера NT/2000/2003/2008 рекомендуется использовать возможности Microsoft WINS. В чистом окружении Samba рекомендуется для WINS использовать только один сервер Samba.

Ниже приведен пример файла /etc/samba/smb.conf, в котором сервер Samba настроен в качестве сервера WINS:

Пример: пример конфигурации сервера WINS

```
[global]
wins support = yes
```

Примечание. Все серверы, включая Samba, для разрешения имен должны подключаться к серверу WINS. Без WINS можно осуществлять просмотр только локальной подсети. И даже если каким-то образом будет доступен список всех машин домена, хосты невозможно будет разрешить без WINS.

16.5.10. Программы в составе Samba

net

```
net <протокол> <функция> <дополнительные_параметры>
<параметры_цели>
```

Утилита net похожа на утилиту net, используемую в Windows и MS-DOS. Первый аргумент служит для указания протокола, используемого при выполнении команды. Значения протокола для указания типа серверного подключения могут быть следующими: `ads`, `rap` или `rpc`. Active Directory использует `ads`, Win9x/NT3 — `rap`, а Windows NT4/2000/2003/2008/2012 — `rpc`. Если протокол не указывается, net автоматически попытается его определить.

В примере ниже показывается список общих ресурсов, доступных для хоста с именем wakko:

```
$ net -l share -S wakko
Password:
Enumerating shared resources (exports) on remote server:
Share name   Type     Description
----------   ----     -----------
data         Disk     Wakko data share
tmp          Disk     Wakko tmp share
IPC$         IPC      IPC Service (Samba Server)
```

```
ADMIN$        IPC       IPC Service (Samba Server)
```

В примере ниже показывается список пользователей Samba для хоста с именем wakko:

```
$ net -l user -S wakko
root password:
User name             Comment
-----------------------------
andriusb              Documentation
joe                   Marketing
lisa                  Sales
```

nmblookup

```
nmblookup <параметры> <имя_netbios>
```

Программа nmblookup разрешает имена NetBIOS в IP-адреса. Программа посылает свои запросы в локальную подсеть до тех пор, пока целевая машина не ответит.

В примере ниже показан IP-адрес трека имени NetBIOS:

```
$ nmblookup trek
querying trek on 10.1.59.255
10.1.56.45 trek<00>
```

pdbedit

```
pdbedit <параметры>
```

Программа pdbedit управляет учетными записями, расположенными в базе данных SAM. Поддерживаются все серверные программы, включая smbpasswd, LDAP и библиотека баз данных tdb.

Ниже показан пример добавления, удаления и получения списка пользователей:

```
$ pdbedit -a kristin
new password:
retype new password:
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-
2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-
2077
Full Name: Home Directory:       \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
```

```
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28
GMT Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT

$ pdbedit -v -L kristin
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-
2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-
2077
Full Name:
Home Directory:       \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28 GMT
Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT

$ pdbedit -L
andriusb:505:
joe:503:
lisa:504:
kristin:506:
~]$ pdbedit -x joe
~]$ pdbedit -L
andriusb:505: lisa:504: kristin:506:

rpcclient
rpcclient <сервер> <параметры>
```

Программа rpcclient вызывает административные команды, используя Microsoft RPC (вызов удаленных процедур Microsoft), предоставляя доступ к графическому интер-

фейсу администратора Windows для управления системами. Чаще всего этот интерфейс используется продвинутыми пользователями, понимающими всю сложность вызова удаленных процедур Microsoft.

smbcacls

```
smbcacls <//сервер/общий_ресурс> <имя_файла> <параметры>
```

Программа smbcacls изменяет списки доступа Windows к файлам и каталогам, являющимися общими ресурсами на сервере Samba или Windows.

smbclient

```
smbclient <//сервер/общий_ресурс> <пароль> <параметры>
```

Программа smbclient — это гибкий клиент UNIX с возможностями, аналогичными возможностям утилиты ftp.

smbcontrol

```
smbcontrol -i <параметры>
smbcontrol <параметры> <место_назначения> <тип_сообщения> <аргу-
менты>
```

Программа smbcontrol посылает контрольные сообщения выполняющимся демонам smbd, nmbd или winbindd. Выполнение `smbcontrol -i` запускает команды интерактивно до тех пор, пока не будет введена пустая строка или символ `q`.

smbpasswd

```
smbpasswd <параметры> <имя_пользователя> <пароль>
```

Программа smbpasswd управляет зашифрованными паролями. Эта программа может выполняться суперпользователем для изменения пароля любого обычного пользователя, а также обычным пользователем для изменения своего собственного пароля Samba.

smbspool

```
smbspool <задача> <пользователь> <название> <копий> <параметры>
<имя_файла>
```

Программа smbspool представляет собой интерфейс печати Samba, совместимый с CUPS. Разработанный в первую очередь для работы с принтерами CUPS, smbspool может работать также и принтерами, не управляемыми CUPS.

smbstatus

```
smbstatus <параметры>
```

Программа smbstatus показывает статус текущих подключений к серверу Samba.

smbtar

```
smbtar <параметры>
```

Программа smbtar создает (а также распаковывает обратно) резервные копии общих файлов и каталогов Windows в ленточных архивах. Программа похожа на утилиту tar, но эти две программы не совместимы между собой.

testparm

```
testparm <параметры> <имя_файла> <имя_хоста адрес_IP>
```

Программа testparm проверяет синтаксис файла /etc/samba/smb.conf. Если файл smb.conf расположен в каталоге по умолчанию (/etc/samba/smb.conf), местоположение

файла указывать не нужно. При указании имени хоста и IP-адреса проверяется правильность конфигурации в файлах hosts.allow и host.deny. После проверки программа testparm также выводит сводку текущего файла smb.conf и выполняемой роли сервера (одиночный, домен и т. д.). Это удобно при отладке, поскольку комментарии отсекаются, и информация предоставляется в сжатом виде, в котором ее могут прочитать опытные администраторы. Например:

```
$ testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[html]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
<enter>
# Global parameters
[global]
   workgroup = MYGROUP
   server string = Samba Server
   security = SHARE
   log file = /var/log/samba/%m.log
   max log size = 50
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
   dns proxy = no
[homes]
   comment = Home Directories
   read only = no
   browseable = no
[printers]
   comment = All Printers
   path = /var/spool/samba
   printable = yes
   browseable = no
[tmp]
   comment = Wakko tmp
   path = /tmp
   guest only = yes
[html]
   comment = Wakko www
   path = /var/www/html
   force user = andriusb
   force group = users
```

```
read only = no
guest only = yes
```

wbinfo

```
wbinfo <параметры>
```

Программа wbinfo показывает информацию, поступающую от демона winbindd. Для работы программы wbinfo необходим запущенный демон winbindd.

Доступная локально документация по Samba

/usr/share/doc/samba-<номер-версии>/ — все дополнительные файлы, входящие в состав пакета программ Samba. Сюда включены все вспомогательные сценарии, примеры конфигурационных файлов, а также документация.

Подробные сведения о конкретных возможностях Samba ищите на страницах следующих руководств:

- smb.conf(5);
- samba(7);
- smbd(8);
- nmbd(8);
- winbindd(8).

17. ИСПОЛЬЗОВАНИЕ ОС РОСА «КОБАЛЬТ» СОВМЕСТНО С ВИРТУАЛЬНОЙ ИНФРАСТРУКТУРОЙ

ОС РОСА «КОБАЛЬТ» может не только устанавливаться на физический компьютер, но и выступать в качестве ОС виртуальной машины.

17.1. Использование ОС РОСА «КОБАЛЬТ» на виртуальных машинах в СУСВ «ROSA Virtualization»

Для корректной работы ОС РОСА «КОБАЛЬТ» в качестве гостевой системы (виртуальной машины ВМ) в системе управления средой виртуализации (СУСВ) «ROSA Virtualization» необходимо после установки ОС провести установку дополнительных пакетов следующей командой:

```
yum install spice-vdagent ovirt-guest-agent
```

Spice-vdagent обеспечит полноценную поддержку удаленного соединения с данной ВМ по протоколу spice, например, масштабирование экрана ВМ.

Ovirt-guest-agent позволит корректно обрабатывать сигналы из СУСВ к ВМ, например, корректное завершение работы, а также позволит более полно отображать информацию о ВМ (FQDN, IP).

17.2. Использование ОС РОСА «КОБАЛЬТ» в качестве тонкого клиента в СУСВ «ROSA Virtualization»

Если ОС РОСА «КОБАЛЬТ» используется для доступа к ВМ, расположенным внутри СУСВ «ROSA Virtualization», для обеспечения полного доступа к удаленному рабочему столу ВМ необходимо установить пакет virt-viewer:

```
yum install virt-viewer
```

18. ПОДКЛЮЧЕНИЕ ОС РОСА «КОБАЛЬТ» К ДОМЕНУ WINDOWS 2008/2012

Для включения станции в домен потребуются права администратора на самой Linux-станции и права администратора домена в AD.

1) Установите следующие пакеты:
   ```
   # yum install authconfig-gtk samba-winbind samba-winbind-clients
   ```
2) Укажите в файле /etc/resolv.conf в качестве первого сервера DNS сервер домена AD либо сервер, отвечающий за DNS-зону домена, к которому будет подключена станция.

   В нашем примере предположим, что сервер AD имеет следующие параметры:
   - IP: 192.168.76.93;
   - FQDN: windc.test.dom.

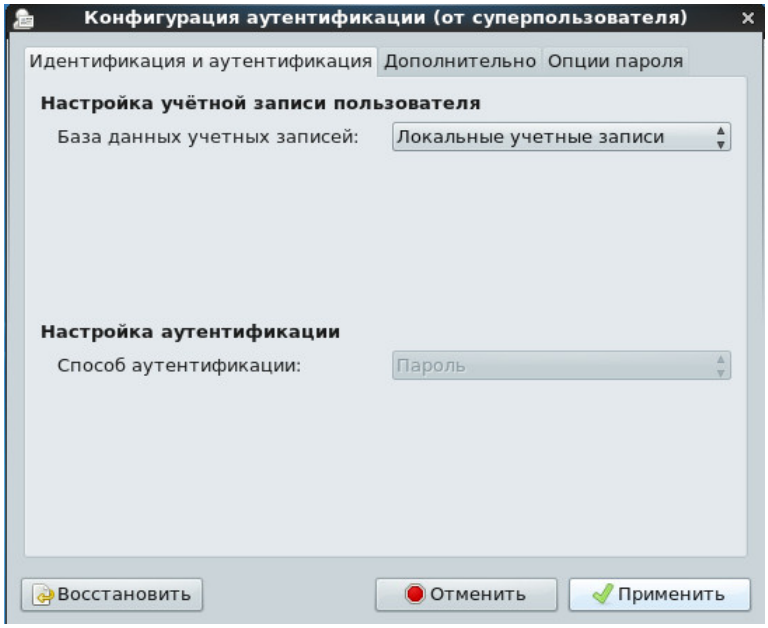
   Таким образом, содержимое файла /etc/resolv.conf должно принять следующий вид:
   ```
   search test.dom
   nameserver 192.168.76.93
   ```

Примечание. Если сетевые параметры для рабочей станции поставляются сервисом DHCP именно он должен выдать правильные параметры, т. к. в противном случае после перезагрузки содержимое файла будет изменено на настройки пришедшие по протоколу DHCP.

3) Проведите синхронизацию времени с контроллером AD в случае, если он является сервером NTP, либо синхронизируйте время с тем же сервером, с которым синхронизируется сам контроллер AD:
   ```
   #ntpdate -u windc.test.dom
   ```
4) Также рекомендуется настроить постоянную синхронизацию времени на рабочей станции, чтобы в процессе эксплуатации не происходило расхождения. По умолчанию в ОС для синхронизации времени используется утилита chrony. Ее конфигурация расположена в файле /etc/chrony.conf. Внесите в данный файл данные своего NTP сервера в опции server, и перезапустите сервис chrony:
   ```
   sudo systemctl restart chronyd
   ```
5) Для синхронизации по времени в момент загрузки ОС внесите данные о вашем NTP сервере в файл /etc/ntp/step-tickers и добавьте в автозагрузку сервис ntpdate:
   ```
   systemctl enable ntpdate
   ```
6) Запустите из командной строки Linux станции графическую утилиту настройки авторизации:
   ```
   authconfig-gtk
   ```

Вам потребуется ввести пароль администратора системы (пользователя, входящего в группу wheel), а если такой отсутствует, то пароль суперпользователя root.

Откроется окно «Конфигурация аутентификации».

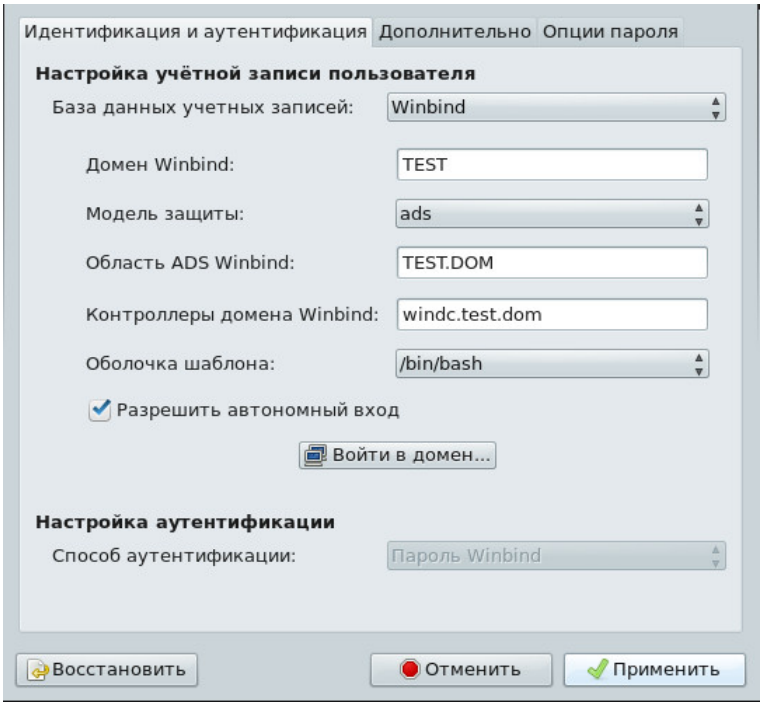


Конфигурация аутентификации

7) В открывшемся окне выберите для поля «База данных учетных записей» вариант «Winbind» и введите следующие данные:

- домен Winbind: TEST;
- модель защиты: ads;
- область ADS Winbind: TEST.DOM;
- контроллер домена Winbind: windc.test.dom;
- оболочка шаблона: /bin/bash.

Также можете поставить галочку в поле «Разрешить автономный вход», чтобы в отсутствие связи с контроллером AD можно было войти в ОС.



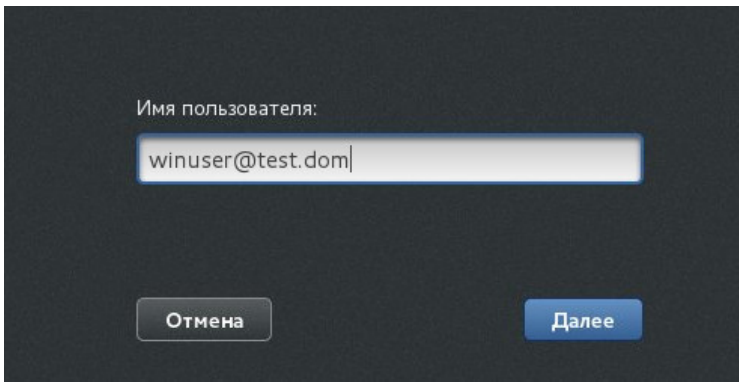
Ввод данных

8) Нажмите на кнопку «Войти в домен» и введите данные администратора домена для подключения к домену AD. Если вам не выдается сообщение об ошибке, значит, вы успешно подключились к домену AD.
9) Нажмите на кнопку [Применить] для завершения процедуры.

Чтобы домашний каталог пользователя AD создавался автоматически при первом входе пользователя в ОС, необходимо в файле /etc/security/pam_winbind.conf установить параметр mkhomedir в yes:

```
mkhomedir = yes
```

10) Перезапустите компьютер.
11) На экране приглашения входа в ОС при выборе пользователя нажмите на указатель «Другой пользователь» и введите имя доменного пользователя в формате <имя_пользователя>@домен, например, winuser@test.dom.



Вход в ОС под доменным пользователем

19. УПРАВЛЕНИЕ СЛУЖБОЙ ИДЕНТИФИКАЦИИ И АУТЕНТИФИКАЦИИ (СИА)

19.1. Домен СИА

Домен управления СИА состоит из группы машин, имеющих общую конфигурацию, общие политики и хранилища удостоверений. Общие параметры делают машины внутри домена совместимыми друг с другом и дают им возможность работать вместе.

С точки зрения СИА, в домен входят следующие типы машин:

- серверы СИА, выполняющие роль контролеров домена;
- клиенты СИА, зарегистрированные на серверах.

Серверы СИА также являются клиентами СИА, зарегистрированными внутри себя: северные машины предоставляют те же самые функции, что и клиенты. СИА поддерживает машины ОС РОСА «КОБАЛЬТ» в качестве серверов и клиентов СИА.

19.2. Описание клиентов и серверов СИА

19.2.1. Серверы СИА — введение

Серверы СИА выполняют роль центральных репозиториев идентификационной информации. На них также располагаются службы, используемые членами домена. СИА предлагает набор инструментов для централизованного управления службами, связанными с СИА: графический интерфейс СИА и утилиты командной строки.

19.2.2. Службы, располагающиеся на серверах СИА

Установка большинства служб не является строго необходимой на сервере СИА. Такие службы, как центр сертификатов, сервер DNS или сервер протокола сетевого времени NTP, можно устанавливать на внешних серверах за пределами домена СИА.

19.2.2.1. Kerberos KDC

СИА использует протокол Kerberos для поддержки единой точки входа. С Kerberos пользователь должен предоставить корректное имя пользователя и пароль только один раз. Далее пользователь получает полный доступ ко всем службам СИА без повторных запросов учетных данных.

19.2.2.2. Сервер каталогов LDAP

Сервер СИА включает в себя экземпляр сервера каталогов LDAP, где хранится вся информация СИА, такая, как данные Kerberos, учетные записи пользователей, записи хостов, служб, политик, DNS и др.

Примечание. В данном руководстве компонент LDAP называется «сервером каталогов».

19.2.2.3. Центр сертификации

В большинстве случаев встроенный центр сертификации (CA) устанавливается вместе с сервером СИА. Если создание и предоставление всех нужных сертификатов выполняется независимо, сервер можно поставить без этого встроенного компонента.

Примечание. В данном руководстве компонент СА называется «системой сертификации», когда речь идет об установленном экземпляре, и «центром сертификации», когда речь идет о службах, предоставляемых этим экземпляром.

19.2.2.4. Система доменных имен (DNS)

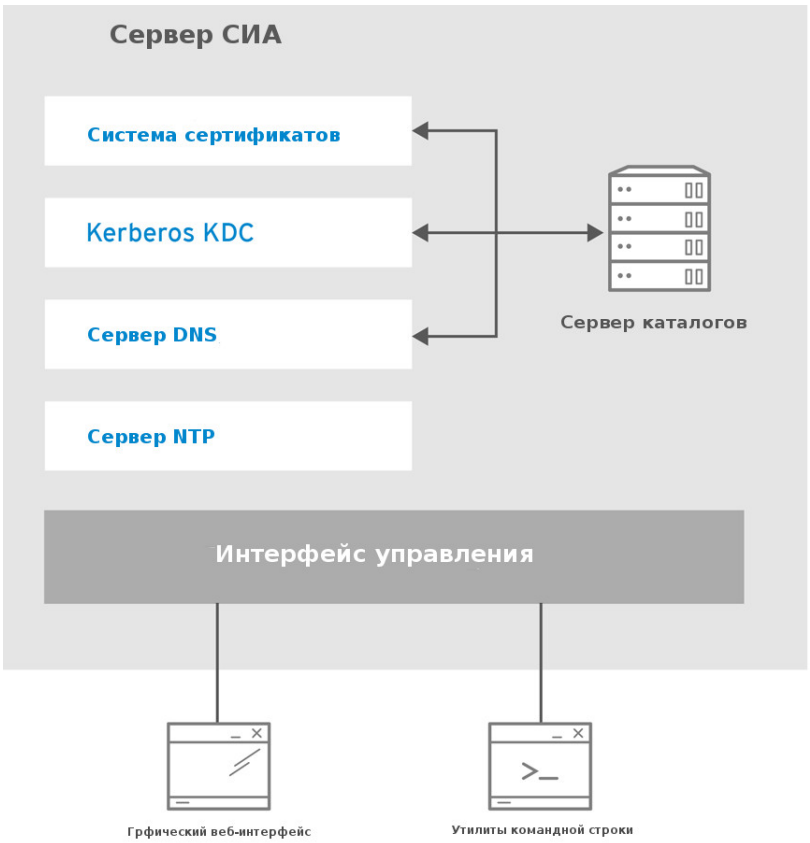
Для динамического обнажения служб СИА использует DNS. Программа установки клиента СИА может использовать данные, полученные от DNS для автоматической настройки машины клиента. После регистрации клиента в домене СИА он использует DNS для нахождения серверов СИА и служб в пределах домена.

19.2.2.5. Протокол сетевого времени NTP

Многим службам необходимо, чтобы системное время на сервере и на клиентских машинах было одинаковым (в пределах некоторого расхождения). Тикеты Kerberos, например, используют метки времени для определения своей действительности и предотвращения атак повторного воспроизведения. Если разница во времени между сервером и клиентом выходит за рамки разрешенного диапазона (по умолчанию 5 мин.), любые тикеты Kerberos становятся недействительными.

По умолчанию для синхронизации часов по сети в СИА используется протокол сетевого времени NTP. С его помощью центральный сервер выполняет роль полномочных часов, а клиенты синхронизируют свое время для совпадения с часами сервера. Во время процесса установки сервер СИА настраивается как сервер NTP для домена СИА.

Схематично взаимодействие сервисов сервера СИА показано на следующем рисунке:



Сервер СИА: службы унификации

19.2.3. Клиенты СИА — введение

Клиенты СИА — это машины, настроенные на работу в домене СИА. Для получения доступа к ресурсам домена эти машины обмениваются данными с серверами СИА. Клиенты, принадлежащие настроенным на серверах доменам Kerberos, получают сертификаты и тикеты, выпущенные серверами, а также используют другие централизованные сервисы для аутентификации и авторизации.

Для активности в составе домена клиенту СИА не требуется специальное клиентское ПО. Все, что нужно, — это корректно настроенные системные параметры определенных служб и библиотек, таких как Kerberos или DNS. Эта конфигурация направляет клиентскую машину на использование служб СИА.

19.2.4. Службы, располагающиеся на клиентах СИА

19.2.4.1. Демон служб системной безопасности (SSSD)

Демон служб системной безопасности (System Security Services Daemon, SSSD) — это клиентское приложение для кэширования регистрационных данных. Использование демона SSSD на клиентских машинах рекомендуется, потому что он упрощает создание необходимых параметров клиента. У демона SSSD также есть и дополнительные полезные функции, например:

- внесетевая аутентификация клиента, которая обеспечивается локальным кэшированием учетных данных, полученных от централизованного идентификатора и из хранилищ аутентификации;
- улучшенная целостность процесса аутентификации, т. к. отсутствует необходимость поддержки и центральной учетной записи и локальной учетной записи пользователя для внесетевой аутентификации;
- интеграция с другими службами, например, с sudo.

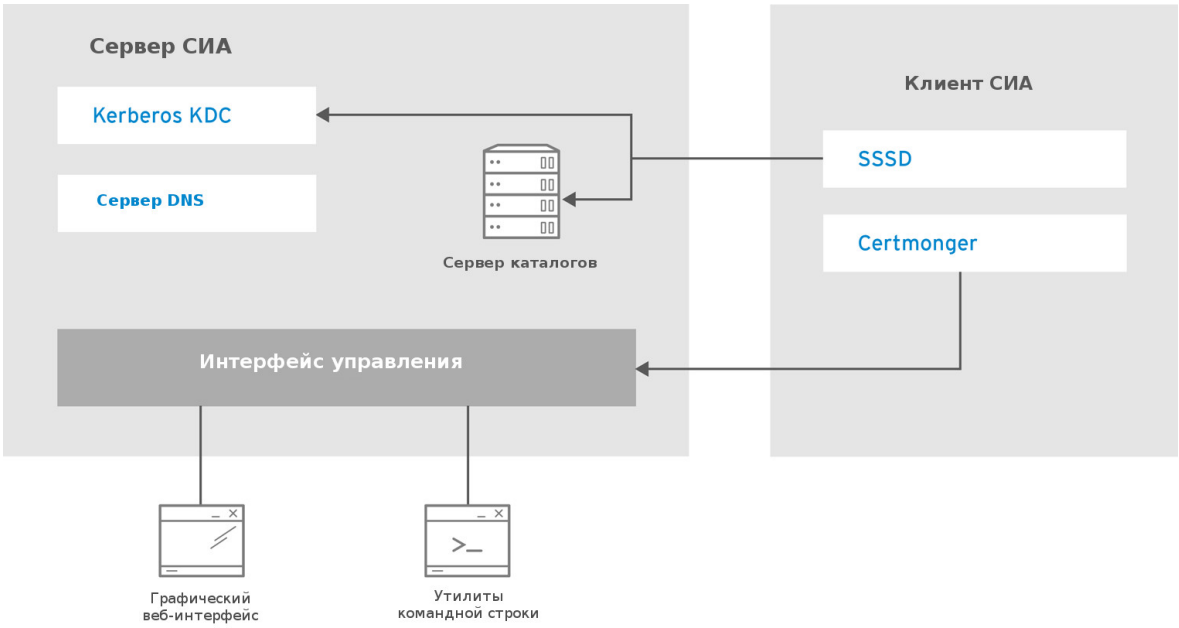
19.2.4.2. Авторизация для контроля доступа со стороны хоста (HBAC)

При использовании SSSD администраторы СИА могут настраивать все параметры идентификации централизованно на сервере СИА. Кэширование дает возможность локальной системе продолжать обычные операции аутентификации и в том случае, если сервер СИА станет недоступен или если клиент отключится от сети.

19.2.4.3. certmonger

Служба certmonger наблюдает за сертификатами и обновляет их на клиенте. Также certmonger может запросить новые сертификаты для служб в системе.

Взаимодействие между службами СИА показано на следующем рисунке:



Взаимодействие между службами СИА

19.3. Установка и удаление сервера СИА

Сервер управления идентификационной информацией (далее — сервер СИА) является контроллером домена СИА. Чтоб настроить сервер СИА, необходимо:

1) Установить нужные пакеты.
2) Настроить машину с помощью установочных сценариев.

Крайне рекомендуется настроить несколько контролеров домена внутри домена для балансировки нагрузки и избыточности. Эти дополнительные серверы являются репликами начального главного сервера СИА.

19.3.1. Предпосылки для установки сервера

19.3.1.1. Аппаратные рекомендации

Наиболее важно выделить правильный объем оперативной памяти. Учитывайте следующие рекомендации:

- для 10 000 пользователей и 100 групп — минимум 2 ГБ оперативной памяти и 1 ГБ файла подкачки;
- для 100 000 пользователей и 50 групп — минимум 16 ГБ оперативной памяти и 4 ГБ файла подкачки.

Примечание. Базовая запись пользователя или простая запись хоста с сертификатом занимает примерно 5–10 КБ дискового пространства.

Для более крупных реализаций более эффективно увеличивать объем памяти, чем размер файла подкачки, т. к. большое количество данных хранится в кэше.

Для улучшения производительности также можно соответственно настроить нижележащий сервер каталогов.

19.3.1.2. Системные требования

В ОС РОСА «КОБАЛЬТ» сервер СИА нужно устанавливать на чистую систему, где отсутствуют настроенные параметры для таких служб, как DNS, Kerberos или сервер каталогов.

Процесс установки сервера СИА переписывает системные файлы для настройки домена СИА. СИА сохраняет резервную копию исходных системных файлов в каталоге /var/lib/ipa/sysrestore/.

Требования демона кэширования службы имен NSCD

На машинах СИА рекомендуется отключать NSCD. Если полное отключение невозможно, активируйте NSCD в тех схемах, где не выполняется кэширование демоном SSSD.

Поскольку, и NSCD и SSSD выполняют кэширование, при одновременном использовании этих служб могут возникнуть проблемы.

Требование поддержки IPv6

Для установки и работы сервера СИА требуется сеть с включенным IPv6. В данном релизе ОС РОСА «КОБАЛЬТ» IPv6 включен по умолчанию.

19.3.2. Настройка имени хоста и DNS

Примечание. Будьте предельно осторожны и убедитесь в том, что доступная служба DNS протестирована и находится в рабочем состоянии, либо такая служба отсутствует или не будет задействована.

Это требование применяется к серверам СИА со встроенной службой DNS, а также к серверам, уставленным без DNS. Записи DNS являются жизненно важными практически для всех доменных функций СИА, включая работу служб каталога LDAP, Kerberos и интеграции Active Directory.

Обратите внимание, что первичный домен DNS и область Kerberos нельзя поменять после установки.

DNS должен быть корректно настроен на хост сервера вне зависимости от того, является ли он встроенным или сторонним.

В случае отсутствия DNS-сервера необходимо добавить запись, описывающую сервер СИА, в локальный файл /etc/hosts:

```
192.168.10.1 ipa.test.dom ipa
```

Управлению СИА требуется отдельный домен DNS для использования с записями служб. Во избежание конфликтов на уровне DNS первичный домен DNS, используемый для СИА, нельзя делать общим для других систем.

Обратите внимание, что имена хостов клиентов СИА не обязательно должны быть частью первичного домена DNS.

19.3.2.1. Проверка имени хоста сервера

Имя хоста должно быть полным доменным именем, например, server.test.dom. Для проверки имени машины используйте утилиту `hostname`:

```
[root@server ~]# hostname
server.test.dom
```

Вывод не должен содержать имя localhost или localhost6.

Примечание. Имя хоста должно быть действительным именем DNS, т. е. разрешается использовать только цифры, символы алфавита и дефис (-). Другие символы в имени хоста, например, нижнее подчеркивание, приведут к сбоям DNS. Кроме этого, имя хоста должно состоять из символов в нижнем регистре, прописные буквы не разрешаются.

Полное имя хоста не должно разрешаться на адрес, замкнутый сам на себя. Он должно разрешаться на публичный IP-адрес машины, а не на 127.0.0.1.

19.3.2.2. Проверка прямой обратной конфигурации DNS

Получите IP-адрес сервера. Команда `ip addr show` покажет и адрес IPv4 и адрес IPv6. Адрес IPv4 показывается в записи, начинающейся со слова `inet`. В следующем примере настроенный адрес IPv4 — 192.0.2.1.

Адрес IPv6 показывается на строке, начинающейся со слова `inet6`. для этого действия имеют значения только адреса со scope global. В примере ниже возвращенный адрес IPv6 равен `2001:DB8::1111`.

```
[root@server ~]# ip addr show
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
   link/ether 00:1a:4a:10:4e:33 brd ff:ff:ff:ff:ff:ff
   inet 192.0.2.1/24 brd 192.0.2.255 scope global dynamic eth0
       valid_lft 106694sec preferred_lft 106694sec
   inet6 2001:DB8::1111/32 scope global dynamic
       valid_lft 2591521sec preferred_lft 604321sec
   inet6 fe80::56ee:75ff:fe2b:def6/64 scope link
          valid_lft forever preferred_lft forever
```

Проверьте настройки прямых запросов DNS с помощью утилиты dig в сочетании с именем хоста.

Выполните команду `dig +short server.test.dom A`. Возвращенный адрес IPv4 должен совпасть с адресом, возвращенным командой ip addr show:

```
[root@server ~]# dig +short server.test.dom A
192.0.2.1
```

Выполните команду `dig +short server.test.dom AAAA`. Если команда вернет адрес, этот адрес должен совпасть с адресом IPv6, возращенным командой `ip addr show`:

```
[root@server ~]# dig +short server.test.dom AAAA
2001:DB8::1111
```

Примечание. Если запись AAAA не вернула никакого вывода, это не значит, что конфигурация неправильная; отсутствие вывода лишь означает, что в DNS для серверной машины не настроен адрес IPv6. Если в сети не планируется использовать адреса IPv6, можно продолжить установку.

Проверьте параметры обратных запросов DNS (записи PTR) с помощью утилиты

dig и IP-адреса. Выполните команду `dig +short -x <адрес_IPv4>`. В выводе должно присутствовать имя хоста, например:

```
[root@server ~]# dig +short -x 192.0.2.1
server.test.dom
```

Если команда `dig +short -x server.test.dom AAAA` на предыдущем шаге вернула адрес IPv6, используйте dig для опроса адреса IPv6. Как и ранее, в выводе команды должно присутствовать имя хоста сервера. Например:

```
[root@server ~]# dig +short -x 2001:DB8::1111
server.test.dom
```

Примечание. Если на предыдущем шаге команда `dig +short server.test.dom AAAA` не вернула никакого адреса IPv6, опрос записи AAAA также ничего не выдаст. В данном случае это нормальное поведение и не указывает на неправильную конфигурацию.

Если имя хоста не показывается или показывается неправильно, даже если команда `dig +short server.test.dom` в предыдущем шаге вернула адрес IP, это указывает на неверную настройку обратного запроса DNS.

19.3.2.3. Проверка перенаправляющих устройств DNS на соответствие стандартам

При настройке СИА совершенно сторонним DNS необходимо проверить, что все средства перенаправления, которые планируется использовать с СИА, соответствуют стандартам EDNS0 и DNSSEC. Для этого нужно просмотреть вывод следующей команды для каждого перенаправляющего сервера в отдельности:

```
$ dig +dnssec @IP_address_of_the_DNS_forwarder . SOA
```

Ожидаемый вывод команды содержит следующую информацию:

- `status: NOERROR`;
- `flags: ra`;
- `EDNS flags: do`;
- запись RRSIG должна присутствовать в разделе `ANSWER`.

Если один из этих элементов отсутствует в выводе, просмотрите документацию устройства перенаправления DNS и проверьте поддержку и активацию стандартов EDNS0 и DNSSEC. Для последних версиях сервера BIND а файле /etc/named.conf должно быть указано `dnssec-enable yes;`.

Ожидаемый ввод может выглядеть таким образом:

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48655
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0,
ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096

;; ANSWER SECTION:
. 31679 IN SOA a.root-servers.net. nstld.verisign-grs.com.
```

```
2015100701 1800 900 604800 86400
. 31679 IN RRSIG SOA 8 0 86400 20151017170000 20151007160000
62530 . GNVz7SQs [...]
```

19.3.2.4. Файл /etc/hosts

Ниже приведен пример правильно настроенного файла /etc/hosts с именем хоста в первой записи и с правильными записями IPv4 и IPv6 для localhost, после которых идет IP-адрес сервера СИА. Обратите внимание, что имя хоста сервера не может быть частью записи localhost.

```
127.0.0.1      localhost.localdomain   localhost
::1                localhost6.localdomain6 localhost6
192.0.2.1      server.test.dom server
2001:DB8::1111 server.test.dom server
```

19.3.3. Требования к портам

Для связи со службами СИА использует определенные порты. Для работы СИА эти порты должны быть открыты и доступны, не должны использоваться другими службами или быть заблокированы межсетевым экраном.

19.3.3.1. Список требуемых портов

Если вы используете сервис блокировки портов (firewalld или iptables), нужно добавить следующие порты или службы в исключение:

Служба	Порты	Протокол
HTTP/HTTPS	80, 443	TCP
LDAP/LDAPS	389, 636	TCP
Kerberos	88, 464	TCP и UDP
DNS	53	TCP и UDP
NTP	123	UDP

Примечание. Не стоит беспокоиться о том, что СИА использует порты 80 и 389.

Порт 80 (HTTP) используется для предоставления откликов OCSP (протокола) и списков аннулирования сертификатов (CRL). Обе программы имеют цифровую подпись и поэтому защищены от атак через посредника (man-in-the-middle).

Порт 389 (LDAP) использует STARTTLS и GSSAPI для шифрования.

Кроме того, СИА может слушать порт 8080, и, в некоторых установках, также и порты 8443 и 749. Тем не менее, эти три порта используются для внутренних подключений, хотя они и открыты, внешний доступ к ним не требуется. Рекомендуется не открывать порты 8080, 8443 и 749 и заблокировать их с помощью сетевого экрана.

19.3.3.2. Список служб firewalld

Имя службы	Подробности см.:
freeipa-ldap	/usr/lib/firewalld/services/freeipa-ldap.xml
freeipa-ldaps	/usr/lib/firewalld/services/freeipa-ldaps.xml
dns	/usr/lib/firewalld/services/dns.xml

19.3.3.3. Открытие необходимых портов

Откройте нужные порты с помощью утилиты firewall-cmd. Выберите одно из следующих решений:

1) Добавление индивидуальных портов в межсетевой экран с помощью команды `firewall-cmd --add-port`. Например, чтобы открыть порты в зоне default, выполните:

```
# firewall-cmd --permanent --add-
port={80/tcp,443/tcp,list_of_ports}
```

2) Добавление служб firewalld в межсетевой экран с помощью команды `firewall-cmd --add-service`. Например, чтобы открыть порты в зоне default, выполните:

```
# firewall-cmd --permanent --add-service={freeipa-
ldap,list_of_services}
```

Перезагрузите конфигурацию firewall-cmd, чтобы изменения вступили в силу немедленно:

```
# firewall-cmd --reload
```

Обратите внимание, что перезапуск службы firewalld на рабочей системе может привести с истечению времени ожидания подключений DNS. При необходимости, чтобы этого избежать, повторите команду без параметра `--permanent` для применения изменений к рабочей системе. Также по желанию можно проверить доступность портов с помощью утилит nc, telnet или nmap или выполнить сканирование портов.

19.3.4. Установка необходимых пакетов для сервера СИА

Установка пакетов для сервера СИА без встроенных служб DNS:

```
# yum install ipa-server
```

Установка пакетов для сервера СИА со встроенными службами DNS:

```
# yum install ipa-server ipa-server-dns
```

Пакеты ipa-server автоматически устанавливают другие пакеты в качестве зависимостей, например:

- 389-ds-base — для службы сервера каталогов LDAP;
- krb5-server — для службы Kerberos.

19.3.5. Установка сервера СИА: начало

Примечание. Процедуры установки и примеры в разделе ниже не являются взаимоисключающими: их можно сочетать для получения необходимого результата. Можно, например, установить сервер со встроенным DNS и с внешним корневым центром сертификации.

Установка и настройка сервера СИА выполняется утилитой ipa-server-install. Перед установкой сервера просмотрите разделы:

- Служба DNS: встроенная или внешняя;
- Выбор параметров Центра сертификации.

Утилита ipa-server-install предоставляет неинтерактивный режим установки с воз-

можностью автоматической установки сервера без участия пользователя. Подробности см. в подразделе «Неинтерактивная установка сервера».

Установочный сценарий ipa-server-install создает файл журнала var/log/ipaserver-install.log. В случае неудачной установки можно просмотреть журнал для нахождения проблемы.

19.3.5.1. Служба DNS: встроенная или внешняя

СИА поддерживает установку сервера как со встроенной службой DNS, так и без нее.

Сервер СИА со встроенной службой DNS

Встроенный сервер DNS, предоставляемый СИА, не предназначен для использования в качестве сервера DNS общего назначения. Он поддерживает только возможности, необходимые для разворачивания и поддержки задач управления идентификационной информацией, и не поддерживает некоторые из продвинутых возможностей DNS.

Для базового использования внутри развернутого сервера СИА настоятельно рекомендуется использовать встроенную службу DNS: когда сервер СИА также управляет и встроенными службами DNS, существует тесная интеграция между DNS и встроенными инструментами СИА, позволяющими автоматизировать некоторые из задач управления записями DNS.

Обратите внимание, что хотя сервер СИА используется как главный сервер DNS, другие внешние серверы DNS также можно использовать как вторичные.

Если, например, в окружении уже используется другой сервер DNS, например, встроенный в Active Directory, то встроенному в СИА серверу можно делегировать только первичный домен СИА. Переносить зоны DNS во встроенный в СИА сервер не требуется.

Сведения об установке сервера СИА со встроенной службой DNS см. в подразделе «Установка сервера СИА со встроенным DNS».

Сервер СИА без встроенной службы DNS

Внешний сервер DNS используется для предоставления служб DNS. Сервер СИА без встроенной службы DNS лучше всего установить в следующих случаях:

- если требуются продвинутые возможности DNS, выходящие за рамки функций, предоставляемых встроенным DNS сервера СИА;
- в окружениях с устоявшейся тщательно настроенной инфраструктурой DNS, позволяющей использовать внешние серверы DNS.

Сведения об установке сервера СИА без встроенной службы DNS см. в подразделе «Установка сервера СИА без встроенного DNS».

Убедитесь, что система отвечает требованиям, описанным в подразделе «Настройка имени хоста и DNS».

Требования к обслуживанию для внешнего или встроенного DNS

При использовании встроенного сервера DNS большинство задач по обслуживанию записей выполняются автоматически. Администратор должен только корректно настроить делегирование с родительского домена на серверы СИА.

Если, например, домен СИА имеет имя ipa.test.dom, оно должно быть корректно де-

легировано из домена test.dom.

Примечание. Делегирование можно проверить с помощью следующей команды:

```
# dig @<IP-адрес>+norecurse +short ipa.test.dom. NS
```

<IP-адрес> — это адрес сервера, управляющего доменом DNS test.dom. Если делегирование было выполнено правильно, эта команда выведет список серверов СИА с установленными серверами DNS.

Если используется внешний сервер DNS, администратор должен:

1) Вручную создать новый домен на сервере DNS.
2) Вручную заполнить новый домен записями из файла зоны, созданного установщиком СИА.
3) Вручную актуализировать записи после установки или удалении реплики, а также делать это после каждого изменения параметров службы, например после настройки доверия для Active Directory.

Предотвращение DNS атак с лавинообразным умножением данных

Параметры по умолчанию сервера DNS, встроенного в СИА, разрешают все клиентам выполнять рекурсивные запросы сервера DNS. Если сервер развернут в сети с недоверенным клиентом, конфигурацию сервера нужно сменить так, чтобы рекурсия была разрешена только авторизованным клиентам.

Для обеспечения этого добавьте записи соответствующего списка ACL в файл /etc/named.conf на сервере. Например:

```
acl authorized { 192.0.2.0/24; 198.51.100.0/24; };
options {
  allow-query { any; };
  allow-recursion { authorized; };
};
```

19.3.5.2. Определение используемой конфигурации центра сертификации

СИА поддерживает установки как со встроенным центром сертификации (CA), так и без него.

Сервер со встроенным центром сертификации

Это конфигурация по умолчанию, и она подходит для большинства случаев. Система сертификатов использует сертификаты, подписанные центром сертификации для создания и подписи сертификатов в домене СИА.

Примечание. Настоятельно рекомендуется установить службы CA на несколько серверов. При установке CA только на один сервер в случае сбоя существует риск потери конфигурации CA без возможности его восстановления.

Подписанный сертификат CA должен быть подписан корневым центром сертификации, являющимся самым высшим звеном в иерархии центров сертификации. Коневой центр может быть как самим сервером СИА, так и внешним центром сертификации.

Центр сертификации СИА является корневым центром

Это конфигурация по умолчанию.

Чтобы установить сервер с такой конфигурацией, см. подраздел «Установка серве-

ра со встроенным DNS» ил раздел «Установка сервера без строенного DNS».

Корневым центром сертификации является внешний CA

Центр сертификации СИА подчиняется внешнему центру. Тем не менее, все сертификаты для домена СИА выдаются экземпляром системы сертификации.

Внешний центр сертификации может быть корпоративным CA или же сторонним, например, Verisign или Thawte. Сертификаты, выданные внутри домена СИА потенциально являются субъектами ограничений, установленных внешним корневым CA для таких атрибутов, как срок действия.

Чтобы установить сервер, использующий внешний центр сертификации, см. подраздел «Установка сервера со внешним центром сертификации в качестве корневого».

Сервер без центра сертификации

Такая конфигурация подходит для очень редких случаев, когда ограничения внутри инфраструктуры не позволяют установить службы сертификации на сервере.

До начала установки необходимо запросить у стороннего центра следующие сертификаты:

1) Сертификат сервера LDAP и закрытый ключ.
2) Сертификат сервера Apache и закрытый ключ.
3) Полную цепочку сертификатов центра сертификации, выдавшего сертификаты для вышеуказанных серверов.

Управление сертификатами без встроенного центра сертификации представляет собой тяжелое административное бремя. Наиболее важные моменты:

1) Создание, загрузка и обновление сертификатов делается вручную.
2) Служба certmonger не используется для отслеживания сертификатов, поэтому предупреждения о приближающемся сроке окончания срока действия сертификатов не поступают.

Чтобы установить сервер без встроенного центра сертификации, см. раздел «Установка без центра сертификации».

19.3.6. Установка сервера со встроенной службой DNS

Примечание. При отсутствии уверенности в том, какую конфигурацию нужно использовать, прочтите подразделы «Служба DNS: встроенная или внешняя» и «Определение используемой конфигурации центра сертификации».

Для процесса установки сервера со встроенным DNS потребуется информация о перенаправляющих средствах DNS.

Поддерживаются следующие параметры перенаправления DNS:

1) Одно или более средств перенаправления (параметр `--forwarder` в неинтерактивной установке).
2) Без перенаправления (параметр `--no-forwarders` в неинтерактивной установке).

19.3.6.1. Обратные зоны DNS

Поддерживаются следующие параметры обратных зон DNS:

1) Автоматическое обнаружение обратных зон, которые необходимо создать в DNS сервера СИА (это параметр по умолчанию в интерактивном режиме установки, и параметр `--auto-reverse` в неинтерактивном режиме).
2) Без автоматического обнаружения обратных зон (параметр `--no-reverse` в интерактивном режиме установки).

   В неинтерактивном режиме также нужно добавить параметр `--setup-dns`.

   Пример: установка сервера со встроенной службой DNS

   В результате этих действий будет установлен сервер:

- со встроенными службами DNS;
- со встроенным центром сертификации в качестве корневого, что является конфигурацией CA по умолчанию.

Запустите утилиту ipa-server-install:

```
# ipa-server-install
```

Сценарий предложит настроить встроенную службу DNS. Введите `yes`.

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```

Сценарий предложит некоторые требуемые параметры. Чтобы принять значения по умолчанию, нажмите <Enter>.

Чтобы предоставить значение, отличное от предложенного по умолчанию, введите его.

```
Server host name [server.test.dom]:
Please confirm the domain name [test.dom]:
Please provide a realm name [test.dom]:
```

Примечание. Крайне рекомендуется, чтобы имя области Kerberos совпадало с доменным именем первичного DNS, написанным в верхнем регистре. Например, если первичный домен DNS называется ipa.test.dom, для названия области Kerberos будет использоваться IPA.test.dom.

Использование различных названий станет препятствием для использования Active Directory, а также может иметь другие негативные последствия.

Введите пароли для суперпользователя сервера каталогов, `cn=Directory Manager` и для пользовательской административной учетной записи сервера СИА.

```
Directory Manager password:
IPA admin password:
The script prompts for DNS forwarders.
Do you want to configure DNS forwarders? [yes]:
```

Для настройки перенаправления DNS введите `yes` и далее следуйте инструкциям.

Процесс установки добавит IP-адреса средств перенаправления в файл /etc/named.conf на установленном сервере СИА.

Параметры по умолчанию см. в описании `--forward-policy` на станице руководства ipa-dns-install(1).

Если перенаправление DNS использовать не нужно, введите `no`.

Сценарий предлагает проверить, нужно ли настроить какие-либо обратные записи DNS (PTR) для адресов IP, связанных с сервером.

```
Do you want to search for missing reverse zones? [yes]:
```

Если в результате поиска будут обнаружены отсутствующие обратные зоны, сценарий спросит, нужно ли создать также и обратные зоны наряду с записями PTR:

```
Do you want to create reverse zone for IP 192.0.2.1 [yes]:
Please specify the reverse zone name [2.0.192.in-addr.arpa.]:
Using reverse zone(s) 2.0.192.in-addr.arpa.
```

Примечание. Управление обратными зонами на СИА является необязательным. Для того можно также использовать внешнюю службу DNS.

Введите `yes` для подтверждения конфигурации сервера.

```
Continue to configure the system with these values? [no]: yes
```

Сценарий установки приступит к настройке сервера. Дождитесь окончания этой операции.

Добавьте делегирование с родительского домена на домен DNS СИА. Если, например, имя домена DNS СИА — ipa.test.dom, добавьте запись сервера доменных имен (NS) в родительский домен test.dom.

Примечание. Этот шаг необходимо повторять каждый раз, когда устанавливается сервер DNS СИА.

Сценарий советует сделать резервную копию сертификата центра сертификации и убедиться в том, что требуемые сетевые порты открыты. Информацию о требованиях СИА к портам и инструкцию о том, как открыть эти порты, см. в подразделе «Требования к портам».

Проверка установленного сервера

1) Выполните аутентификацию в области Kerberos с использованием учетной записи администратора. Это даст уверенность в том, что учетная запись администратора корректно настроена и область Kerberos доступна.

   ```
   # kinit admin
   ```

2) Запустите команду, например, ipa user-find. На новом сервере команда выведет только одного настроенного пользователя: admin.

   ```
   # ipa user-find admin
   --------------
   1 user matched
   --------------
   User login: admin
   Last name: Administrator
   Home directory: /home/admin
   Login shell: /bin/bash
   UID: 939000000
   GID: 939000000
   Account disabled: False
   Password: True
   ```

```
Kerberos keys available: True
----------------------------
Number of entries returned 1
----------------------------
```

19.3.7. Установка сервера без встроенных служб DNS

Примечание. При отсутствии уверенности в том, какую конфигурацию нужно использовать, прочтите подразделы «Служба DNS: встроенная или внешняя» и «Определение используемой конфигурации центра сертификации».

Чтобы установить сервер без встроенного DNS, запустите утилиту ipa-server-install без параметров, имеющих отношение к DNS.

Пример: установка сервера без встроенной службы DNS

В результате этих действий будет установлен сервер:

- без встроенных служб DNS;
- со встроенным центром сертификации в качестве корневого, что является параметром CA по умолчанию.

Запустите утилиту ipa-server-install:

```
# ipa-server-install
The script prompts to configure an integrated DNS service. Press
Enter to select the default no option.
```

Сценарий предложит настроить встроенную службу DNS. Нажмите <Enter>, чтобы указать ответ `no`.

```
Do you want to configure integrated DNS (BIND)? [no]:
```

Сценарий предложит ввести некоторые требуемые параметры.

Чтобы принять значения по умолчанию в квадратных скобках, нажмите <Enter>. Чтобы предоставить значение, отличное от предлагаемого, введите нужное значение.

```
Server host name [server.test.dom]:
Please confirm the domain name [test.dom]:
Please provide a realm name [test.dom]:
```

Примечание. Крайне рекомендуется, чтобы имя области Kerberos совпадало с доменным именем первичного DNS, написанным в верхнем регистре. Например: если первичный домен DNS называется ipa.test.dom, для названия области Kerberos будет использоваться IPA.test.dom.

Использование различных названий станет препятствием для использования Active Directory, а также может иметь другие негативные последствия.

Введите пароли для суперпользователя сервера каталогов, `cn=Directory Manager` и для пользовательской административной учетной записи сервера СИА.

```
Directory Manager password:
IPA admin password:
```

Введите `yes` для подтверждения конфигурации сервера.

```
Continue to configure the system with these values? [no]: yes
```

Сценарий установки приступит к настройке сервера. Дождитесь окончания этой операции.

Сценарий установки создаст файл с записями о ресурсах DNS: файл /tmp/ipa.system.records.UFRPto.db, примерное содержимое которого показано ниже. Добавьте эти записи к существующим внешним серверам DNS. Процесс актуализации записей DNS отличается в зависимости от конкретных параметров DNS.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

Примечание. Установка сервера не считается законченной до тех пор, пока записи DNS не будут добавлены на существующие серверы DNS.

Сценарий советует сделать резервную копию сертификата центра сертификации и убедиться в том, что требуемые сетевые порты открыты. Информацию о требованиях СИА к портам и инструкцию о том, как открыть эти порты, см. в подразделе «Требования к портам».

19.3.7.1. Проверка установленного сервера

Выполните аутентификацию в области Kerberos с использованием учетной записи администратора. Это даст уверенность в том, что учетная запись администратора корректно настроена и область Kerberos доступна.

```
# kinit admin
```

Запустите команду, например, `ipa user-find`. На новом сервере команда выведет только одного настроенного пользователя: admin.

```
# ipa user-find admin
--------------
1 user matched
--------------
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 939000000
GID: 939000000
Account disabled: False
Password: True
Kerberos keys available: True
----------------------------
Number of entries returned 1
----------------------------
```

19.3.8. Установка сервера с внешним центром сертификации в качестве корневого

Примечание. При отсутствии уверенности в том, какую конфигурацию нужно использовать, прочтите разделы «Служба DNS: встроенная или внешняя» и «Определение используемой конфигурации центра сертификации».

Чтобы установить сервер и соединить его со внешним центром сертификации, играющим роль корневого, передайте следующие параметры сценарию установки ipa-server-install:

- параметр `--external-ca` указывает на то, что нужно использовать внешний CA;
- параметр `--external-ca-type` указывает тип внешнего CA. Подробности см. на странице руководства ipa-server-install(1).

Во всем остальном процедура установки аналогична той, что описана в подразделе «Установка сервера со встроенными службами DNS» или в подразделе «Установка сервера без встроенной службы DNS».

Во время настройки экземпляра системы сертификации утилита выводит местоположение запроса на подпись сертификата (CSR): /root/ipa.csr:

```
...
Configuring certificate server (pki-tomcatd): Estimated time 3
minutes 30 seconds
  [1/8]: creating certificate server user
  [2/8]: configuring certificate server instance
```

Следующий шаг — получение /root/ipa.csr, подписанного центром сертификации сервера, и повторно запустить /sbin/ipa-server-install в такой форме:

```
/sbin/ipa-server-install --external-cert-file=/путь/до/подписан-
ного_сертификата --external-cert-file=</путь/до/внешнего_сертифи-
ката_ЦС>
```

После этого предоставьте CSR, расположенный по адресу /root/ipa.csr, внешнему центру сертификации. Этот процесс отличается в зависимости от службы, используемой в качестве внешнего центр сертификации.

Примечание. Запросить соответствующее расширение сертификата может оказаться важно. Сертификат, созданный для сервера СИА, должен быть действительным сертификатом центра сертификации. Для этого должно быть выполнено одно из двух условий:

1) Параметр Basic Constraint должен иметь значение `CA=true`.
2) Для разрешения подписи сертификатов расширение Key Usage Extension должно быть настроено на сертификат подписи.

Получите выпущенный сертификат и цепочку сертификатов центра сертификации для выпускающего центра сертификации в базовом blob-объекте с 64-битным кодированием (файл PEM или сертификат Base_64 от центра сертификации Windows). Опять же, для каждой службы сертификации процесс может отличаться. Обычно администратор скачивает нужный сертификат по ссылке на веб-странице или в почтовом уведомлении.

Примечание. Убедитесь в том, что получаете полную цепочку сертификатов для

центра сертификации, а не только сертификат ЦС.

Повторно запустите ipa-server-install, на этот раз указав местоположения и названия свежевыпущенного сертификата ЦС и файлы цепочки ЦС. Например:

```
# ipa-server-install --external-cert-
file=/tmp/servercert20110601.pem --external-cert-
file=/tmp/cacert.pem
```

Примечание. Команда `ipa-server-install --external-ca` может иногда закончиться неудачей со следующей ошибкой:

```
ipa         : CRITICAL failed to configure ca instance Command
'/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned
non-zero exit status 1
Configuration of CA failed
```

Этот сбой случается при настроенных переменных окружения *_proxy. Решение для этой проблемы см. в подразделе «Сбой установки внешнего центра сертификации».

19.3.9. Установка без центра сертификации

Примечание. При отсутствии уверенности в том, какую конфигурацию нужно использовать, прочтите разделы «Служба DNS: встроенная или внешняя» и «Определение используемой конфигурации центра сертификации».

Чтобы установить сервер без центра сертификации, необходимо вручную предоставить требуемые сертификаты, предоставив параметры для утилиты ipa-server-install. Во всем остальном процедура установки аналогична той, что описана в подразделе «Установка сервера со встроенными службами DNS» или в подразделе «Установка сервера без встроенной службы DNS».

Примечание. Сервер или реплику нельзя установить с использованием самоподписанных сторонних сертификатов сервера.

Чтобы предоставить сертификат сервера LDAP и закрытый ключ:

- `--dirsrv-cert-file` для файлов сертификата и закрытого ключа сертификата сервера LDAP;
- `--dirsrv-pin` для пароля доступа к закрытому ключу в файлах, указанных в `--dirsrv-cert-file`.

Чтобы предоставить сертификат сервера Apache и закрытый ключ:

- `--http-cert-file` для файлов сертификата и закрытого ключа сертификата сервера Apache;
- `--http-pin` для пароля доступа к закрытому ключу в файлах, указанных в `--http-cert-file`.

Чтобы предоставить полную цепочку сертификатов центра сертификации, выдавшего сертификаты для серверов LDAP и Apache:

`--dirsrv-cert-file` и `--http-cert-file` для файлов сертификата полной цепочки сертификатов ЦС или части ее.

Эти параметры можно добавлять несколько раз. Принимаются:

- файлы сертификатов X.509, зашифрованные в кодировке DER и PEM;
- файлы закрытых ключей PKCS#1 и PKCS#8;
- файлы цепочки PKCS#7;
- файлы PKCS#12.

Файлы, указанные параметрами `--dirsrv-cert-file` и `--http-cert-file`, должны содержать только один сертификат сервера и только один закрытый ключ. Содержимое файлов, предоставленных параметрами `--dirsrv-cert-file` и `--http-cert-file`, часто идентично.

При необходимости добавьте параметр `--ca-cert-file` для файлов, заканчивающих цепочку сертификатов центра сертификации.

Этот параметр можно указывать несколько раз. Принимаются:

- файлы сертификатов X.509, зашифрованные в кодировке DER и PEM;
- файлы цепочки сертификатов PKCS#7;

Файлы, предоставленные с ключом `--dirsrv-cert-file` и `--http-cert-file`, в сочетании с файлами, указанными с ключом `--ca-cert-file`, должны содержать полную цепочку сертификатов центра сертификации, выдавшего сертификаты серверов LDAP и Apache.

Например:

```
[root@server ~]# ipa-server-install \
    --http-cert-file /tmp/server.crt \
    --http-cert-file /tmp/server.key \
    --http-pin secret \
    --dirsrv-cert-file /tmp/server.crt \
    --dirsrv-cert-file /tmp/server.key \
    --dirsrv-pin secret \
    --ca-cert-file ca.crt
```

Обратите внимание, что параметры командной строки в этом разделе несовместимы с ключом `--external-ca`.

Примечание. В предыдущих версиях управления идентификационной информацией для указания файла PEM сертификата корневого центра сертификации использовался параметр `--root-ca-file`. Теперь это больше не нужно, т. к. доверенный центр сертификации всегда является инстанцией, выдавшей сертификаты серверов DS и HTTP. СИА теперь автоматически распознает сертификат корневого центра среди сертификатов, указанных с помощью параметров `--dirsrv-cert-file`, `--http-cert-file` и —`ca-cert-file`.

19.3.10. Установка сервера без участия администратора (неинтерактивная установка)

Примечание. При отсутствии уверенности в том, какую конфигурацию DNS или CA нужно использовать, прочтите разделы «Служба DNS: встроенная или внешняя» и «Определение используемой конфигурации центра сертификации».

Минимальный набор параметров, который нужно предоставить для неинтерактивной установки:

- `--ds-password` для указания пароля Directory Manager (DM), суперпользователя сервера каталогов;
- `--admin-password` для указания пароля пользователя admin, администратора СИА;
- `--realm` для указания имени области Kerberos;
- `--unattended` для указания процессу установки выбрать параметры по умолчанию для имени хоста и имени домена.

Коме того, при необходимости можно указать конкретные значения для следующих параметров:

- `--hostname` для имени хоста сервера;
- `--domain` для имени домена.

Примечание. Крайне рекомендуется, чтобы имя области Kerberos совпадало с доменным именем первичного DNS, написанным в верхнем регистре. Например: если первичный домен DNS называется ipa.test.dom, то для названия области Kerberos будет использоваться IPA.test.dom.

Использование различных названий станет препятствием для использования Active Directory, а также может иметь другие негативные последствия.

Чтобы получить полный список параметров, принимаемых командой `ipa-server-install`, выполните команду `ipa-server-install --help`.

Пример: базовая установка без участия администратора (неинтерактивная)

Выполните команду ipa-server-install с указанием нужных параметров. Следующая команда, например, устанавливает сервер без встроенной службы DNS и со встроенным центром сертификации:

```
# ipa-server-install --realm test.dom --ds-password DM_password
--admin-password admin_password --unattended
```

Сценарий установки приступит к настройке сервера. Дождитесь окончания этой операции.

Сценарий установки создаст файл с записями ресурсов DNS: файл /tmp/ipa.system.records.UFRPto.db, примерное содержимое которого приведено ниже. Добавьте эти записи к существующим внешним серверам DNS. Процесс актуализации записей DNS зависит от конкретных параметров DNS.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

Примечание. Установка сервера не считается законченной до тех пор, пока записи DNS не будут добавлены в существующие серверы DNS.

Сценарий советует сделать резервную копию сертификата центра сертификации и убедиться в том, что требуемые сетевые порты открыты. Информацию о требованиях СИА к портам и инструкцию о том, как открыть эти порты, см. в подразделе «Требования к портам».

19.3.10.1. Проверка установленного сервера

Выполните аутентификацию в области Kerberos с использованием учетной записи администратора. Это даст уверенность в том, что учетная запись администратора корректно настроена и область Kerberos доступна.

```
# kinit admin
```

Запустите команду, например, `ipa user-find`. На новом сервере команда выведет только одного настроенного пользователя: admin.

```
# ipa user-find admin
--------------
1 user matched
--------------
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 939000000
GID: 939000000
Account disabled: False
Password: True
Kerberos keys available: True
----------------------------
Number of entries returned 1
```

19.3.11. Удаление сервера СИА

Чтобы удалить server.test.dom:

Запустите на другом сервере команду ipa server-del для удаления server.test.dom из топологии:

```
[root@another_server ~]# ipa server-del server.test.dom
```

На сервере server.test.dom выполните следующую команду:

```
[root@server ~]# ipa-server-install --uninstall
```

Убедитесь в том, что все записи DNS, указывающие на server.test.dom, удалены из зон DNS. Это делается при любой конфигурации DNS: со встроенной службой, управляемой СИА, или с внешним сервером.

19.3.12. Переименование сервера

Невозможно сменить имя хоста сервера СИА после того, как оно было настроено. Тем не менее, сервер можно заменить репликой с другим именем.

1) Создайте новую реплику сервера с центром сертификатов и с новым именем хоста

или адресом IP.

2) Остановите работу начального экземпляра сервера СИА:

```
[root@old_server ~]# ipactl stop
```

3) Проверьте, что все реплики и клиенты работают, как и прежде.
4) Удалите начальный сервер СИА, как описано в разделе «**Удаление сервера СИА**».

19.4. Установка и удаление клиентов СИА

В данном подразделе объясняется, как настроить систему для присоединение к домену СИА в качестве клиентской машины, зарегистрированной на сервере.

19.4.1. Предпосылки для установки клиента

19.4.1.1. Требования к DNS

Необходимо выполнить корректное делегирование DNS. Подробности о требованиях к DNS на сервере СИА см. в подразделе «Настройка имени хоста и DNS».

Убедитесь что подключаемый хост имеет уникальное имя в домене (не localhost и не совпадающее с другим членом домена). Имя указывается в файле /etc/hostname. В файле /etc/resolv.conf указан DNS-сервер, отвечающий за данный домен.

19.4.1.2. Требования к портам

Для обмена данными со своими службами клиенты СИА подключаются к некоторому количеству портов на серверах СИА. Чтобы сервер СИА могли работать, эти порты необходимо открыть. Подробнее о том, какие порты необходимы серверу СИА, см. в подразделе «Требования к портам».

На клиентах откройте эти порты в исходящем направлении. Если используется межсетевой экран, не фильтрующий исходящие пакеты, такой, как firewalld, в исходящем направлении порты уже будут доступны.

19.4.1.3. Требования для демона NSCD (Service Cache Daemon)

Рекомендуется отключить работу NSCD на машинах СИА. Если полностью отключить работу NSCD невозможно, используйте NSCD в тех схемах, где SSSD не выполняет кэширование.

Как служба NSCD, так и служба SSSD выполняют кэширование, поэтому при одновременном использовании этих двух служб могут возникнуть проблемы.

19.4.1.4. Пакеты, необходимые для установки клиента

Установите пакет ipa-client:

```
# yum install ipa-client
```

Пакет ipa-client автоматически установит другие необходимые пакеты в виде зависимостей, например, пакеты демона служб безопасности системы SSSD (System Security Services Daemon).

Для возможности управления доменом СИА с клиентской машины также установите пакет ipa-admintools. В пакете находятся консольные утилиты для администрирования СИА. Если клиентскую машину планируется использовать как обычного клиента, ipa-

admintools не нужны.

19.4.2. Установка клиента

Утилита ipa-client-install устанавливает и настраивает клиент СИА. В процессе установки администратор должен предоставить данные учетных записей, которые можно использовать для регистрации клиента на сервере. Поддерживаются следующие способы аутентификации:

Учетные данные пользователя, имеющего полномочия регистрации клиента на сервере, например, пользователя admin

По умолчанию ipa-client-install ожидает этого параметра. Пример см. в подразделе «Интерактивная установка клиента».

Для явного предоставления данных учетной записи утилите ipa-client-install используйте параметры `--principal` и `--password`.

Случайный одноразовый пароль, предварительно созданный на сервере

Чтобы использовать этот метод аутентификации, добавьте параметр `--random` для команды ipa-client-install. Смотрите пример «Неинтерактивная установка клиента с использованием случайного пароля».

Принципал из предыдущей регистрации на сервере

Чтобы использовать этот метод аутентификации, добавьте параметр `--keytab` для команды ipa-client-install. Подробности см. в разделе «Повторная регистрация клиента в домене СИА» и на странице руководства ipa-client-install(1).

В разделах ниже описываются базовые сценарии установки. Более подробную информацию об использовании утилиты ipa-client-install и полный список поддерживаемых передаваемых параметров см. на странице руководства ipa-client-install(1).

19.4.2.1. Интерактивная установка клиента

При установке с помощью этого способа администратор должен по запросу сценария установки указывать необходимые данные пользователя, уполномоченного регистрировать клиента в домене, например, пользователя admin.

Запустите утилиту ipa-client-install.

При наличии одного из нижеуказанных условий добавьте параметр `--enable-dns-updates` для актуализации записей DNS с использованием адреса IP клиентской машины:

- сервер СИА, на котором будет зарегистрирован клиент, был установлен со встроенной службой DNS;
- сервер DNS в сети принимает записи DNS, актуализированные с помощью протокола GSS-TSIG.

Для отключения хранения паролей Kerberos в кэше SSSD добавьте параметр `--no-krb5-offline-passwords`.

Сценарий установки попытается получить все необходимые параметры автоматически.

Если зона DNS и записи SRV настроены корректно, сценарий автоматически обнаружит все требуемые значения и выведет их на экран. Для подтверждения введите `yes`.

```
Client hostname: client.test.dom
Realm: test.dom
DNS Domain: test.dom
IPA Server: server.test.dom
BaseDN: dc=example,dc=com

Continue to configure the system with these values? [no]: yes
```

Если требуется установить систему с другими значениями, отмените текущую установку и затем снова запустите ipa-client-install, указав нужные значения с помощью параметров.

Подробности см. в разделе DNS Autodiscovery страницы руководства ipa-client-install(1).

Если сценарий не может получить какие-либо значения автоматически, выводится их запрос.

Примечание. Имя сервера должно быть действительным именем DNS, т. е. разрешается использовать только цифры, символы алфавита и дефис (-). Другие символы в имени хоста, например, нижнее подчеркивание, приведут к сбоям DNS. Кроме того, разрешается использовать только нижний регистр.

Сценарий выведет запрос данных пользователя, который будет использован при регистрации клиента на сервере. По умолчанию это пользователь admin:

```
User authorized to enroll computers: admin
Password for admin@test.dom
```

Сценарий установки начнет настройку клиента. Дождитесь завершения операций.

```
Client configuration complete.
```

19.4.2.2. Неинтерактивная установка клиента

Для неинтерактивной установки (без участия администратора) утилите ipa-client-install необходимо сразу предоставить всю требуемую информацию с помощью консольных параметров. Минимальные сведения, требуемые для неинтерактивной установки:

1) Параметры для указания данных учетной записи, используемой для регистрации клиента на сервере, подробности см. в разделе «Установка клиента».
2) Параметр `--unattended` для запуска установки без выводов запросов информации.

Если зона DNS и записи SRV настроены в системе корректно, сценарий автоматически обнаружит другие требуемые значения. В противном случае их необходимо предоставить с помощью консольных параметров.

3) Параметр `--hostname` для указания статического имени хоста клиентской машины.

Примечание. Полное имя домена должно быть действительным именем DNS, т. е. разрешается использовать только цифры, символы алфавита и дефисы (-). Другие символы в имени хоста, например, нижнее подчеркивание, приведут к сбоям DNS. Кроме того,

разрешается использовать только нижний регистр.

4) Параметр `--server` для указания имя хоста сервера СИА, на котором будет зарегистрирован клиент.
5) Параметр `--domain` для указания доменного имени DNS сервера СИА, на котором будет зарегистрирован клиент.
6) Параметр `--realm` для указания имени области Kerberos.

При наличии одного из нижеуказанных условий добавьте параметр `--enable-dns-updates` для актуализации записей DNS с использованием адреса IP клиентской машины:

- сервер СИА, на котором будет зарегистрирован клиент, был установлен со встроенной службой DNS;
- сервер DNS в сети принимает записи DNS, актуализированные с помощью протокола GSS-TSIG.

Для отключения хранения паролей Kerberos в кэше SSSD добавьте параметр `--no-krb5-offline-passwords`.

Полный список параметров, принимаемых утилитой ipa-client-install, см. на странице руководства ipa-client-install(1).

Пример: неинтерактивная установка клиента со случайным паролем

В данном примере клиент устанавливается без ввода информации со стороны администратора. На сервере, используемом доля авторизации регистрации, предварительно создается случайный одноразовый пароль.

На существующем сервере:

1) Войдите в систему с правами администратора:
```
$ kinit admin
```
2) Добавьте новую машину как хост СИА. Для создания случайного пароля используйте параметр `--random`:
```
$ ipa host-add client.test.dom --random
--------------------------------------------------
Added host "client.test.dom"
--------------------------------------------------
  Host name: client.test.dom
  Random password: W5YpARl=7M.n
  Password: True
  Keytab: False
  Managed by: server.test.dom
```

Созданный пароль станет недействительным после его использования для регистрации машины в домене СИА. По завершении регистрации пароль будет заменен корректной таблицей ключей хоста.

На машине, где планируется установить клиент, запустите команду ipa-client-install со следующими параметрами:

- `--password` для создания случайного пароля из вывода ipa host-add;

- `--unattended` для запуска установки без запросов информации.

Примечание. Пароль часто содержит специальные символы, поэтому заключите его в одинарные кавычки (').

Если зона DNS и записи SRV настроены в системе корректно, сценарий автоматически обнаружит другие требуемые значения. В противном случае их необходимо предоставить с помощью параметров.

Например:

```
# ipa-client-install --password 'W5YpARl=7M.n' --domain d-
abc.idm.lab.eng.brq.redhat.com --server vm-058-
105.abc.idm.lab.eng.brq.redhat.com --unattended
```

Запустите утилиту ipa-client-automount, которая автоматически настроит NFS для СИА.

19.4.3. Настройка клиента СИА с использованием Kickstart

Регистрация Kickstart автоматически добавляет новую систему в домен СИА во время установки ОС РОСА «КОБАЛЬТ».

Подготовка к установке клиента с помощью Kickstart включает в себя следующие шаги:

1) Предварительное создание записи хоста клиента на сервере СИА.
2) Создание файла Kickstart для клиента.

19.4.3.1. Предварительное создание записи хоста клиента на сервере СИА

1) Войдите в систему с правами пользователя admin:
   ```
   $ kinit admin
   ```
2) Создайте запись для хоста на сервере СИА и установите для этой записи временный пароль:
   ```
   $ ipa host-add client.test.dom --password=secret
   ```

Kickstart использует этот пароль для аутентификации во время установки клиента, и его срок действия истекает после первой попытки аутентификации. После успешной установки клиента для аутентификации используется его таблица ключей.

19.4.3.2. Создание файла Kickstart для клиента

Файл Kickstart, используемый для настройки клиента, должен иметь следующее содержимое:

1) Пакет ipa-client в списке устанавливаемых пакетов:
   ```
   %packages
   @ X Window System
   @ Desktop
   @ Sound and Video
   ipa-client
   ...
   ```
2) Действия, выполняемые после установки, которые:

- обеспечивают создание ключей SSH до регистрации клиента на сервере;
- запускают утилиту ipa-client-install, для которой указывается:

а) вся информация, необходимая для получения доступа и настройки служб домена СИА

б) пароль, установленный во время предварительного создания хоста клиента на сервере СИА, в подразделе «Предварительное создание записи хоста клиента на сервере СИА».

Например:

```
%post --log=/root/ks-post.log

# Generate SSH keys to ensure that ipa-client-install uploads
them to the IdM server
/usr/sbin/sshd-keygen

# Run the client install script
/usr/sbin/ipa-client-install --hostname=client.test.dom
--domain=test.dom --enable-dns-updates --mkhomedir -w secret
--realm=test.dom --server=server.test.dom
```

Для неинтерактивной установки также добавьте параметр `--unattended`.

Чтобы разрешить сценарию установки запросить сертификат для машины, добавьте параметр `--request-cert` команде ipa-client-install.

В окружении kickstart chroot настройте адрес системной шины на /dev/null для утилит getcert и ipa-client-install. Для этого добавьте следующие записи в файл инструкций, выполняемых после установки до действий ipa-client-install:

```
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null getcert list
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null ipa-client-
install
```

Примечание. Рекомендуется не запускать службу sshd до выполнения регистрации kickstart. Несмотря на то, что запуск sshd до регистрации клиента на сервере создает ключи SSH автоматически, использование вышеприведенного скрипта считается предпочитаемым решением.

19.4.4. Возможные действия после установки клиента

19.4.4.1. Удаление параметров, существовавших до настройки управления идентификационной информацией

Сценарий ipa-client-install не удаляет ранее существовавшие конфигурации LDAP и SSSD из файлов /etc/openldap/ldap.conf и /etc/sssd/sssd.conf. Если до установки клиента параметры в этих файлах изменялись, сценарий добавит новые значения для клиента, но в закомментированный виде. Например:

```
BASE    dc=example,dc=com
URI     ldap://ldap.test.dom
```

```
#URI ldaps://server.test.dom # modified by IPA
#BASE dc=ipa,dc=example,dc=com # modified by IPA
```

Чтобы применить новые параметры СИА:

1) Откройте файлы /etc/openldap/ldap.conf и /etc/sssd/sssd.conf.
2) Удалите предыдущие параметры.
3) Удалите комментарии с новых параметров СИА.

Для серверных процессов, зависящих от системных параметров LDAP, может понадобиться повторный запуск для применения параметров. Приложения, использующие библиотеки openldap, обычно при запуске выполняют импорт конфигурации.

19.4.4.2. Тестирование нового клиента

Проверьте, может ли клиент получить информацию о пользователях, настроенную на сервере, например, о пользователе по умолчанию admin:

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins)
groups=1254400000(admins)
```

19.4.4.3. Удаление клиента

Клиент удаляется из домена СИА вместе со всеми параметрами, имеющими отношение к службам СИА, таким, как SSSD. Это позволяет восстановить предыдущие параметры на клиентской машине.

Команда для удаления клиента:

```
# ipa-client-install --uninstall
```

Вручную удалите записи DNS клиентского хоста с сервера.

19.4.5. Повторная регистрация клиента в домене СИА

Если виртуальная машина клиента был удалена, но ее таблица ключей все еще существует, клиента можно повторно зарегистрировать на сервере:

1) Интерактивно с использованием данных учетной записи администратора. См. подраздел «интерактивная повторная регистрация клиента на сервере с использованием учетной записи администратора».
2) Неинтерактивно с использованием файла таблицы ключей, для которого ранее была сделана резервная копия. См. подраздел «Неинтерактивная повторная регистрация клиента с использованием таблицы ключей клиента».

Примечание. Повторно зарегистрировать можно только тех клиентов, для которых доменная запись все еще активна. Если клиент был удален (с помощью `ipa-client-install --uninstall`) или запись о его хосте была отключена (с помощью ipa host-disable), провести его повторную регистрацию невозможно.

Во время повторной регистрации СИА выполняет следующие действия:

1) Отзыв исходного сертификата хоста.
2) Создание нового сертификата хоста.
3) Создание новых ключей SSH.

4) Создание новой таблицы ключей.

19.4.5.1. Интерактивная повторная регистрация клиента на сервере с использованием учетной записи администратора

Повторно создайте машину клиента с тем же именем хоста.

На клиентской машине выполните следующую команду:

```
# ipa-client-install --force-join
```

Сценарий запросит о пользователе, учетная запись которого будет использована для регистрации клиента. По умолчанию это пользователь admin:

```
User authorized to enroll computers: admin
Password for admin@test.dom
```

19.4.5.2. Неинтерактивная повторная регистрация клиента с использованием таблицы ключей клиента

Повторная регистрация с использованием таблицы ключей подходит для автоматической установки или в других ситуациях, когда использование пароля администратора неоправданно.

1) Создайте резервную копию исходного клиентского файла таблицы ключей, например, в каталоге /tmp или /root.
2) Повторно создайте клиентскую машину с тем же именем хоста.
3) Повторно выполните регистрацию клиента на сервере и укажите местоположение таблицы ключей с помощью параметра `--keytab`:

```
# ipa-client-install --keytab /tmp/krb5.keytab
```

Примечание. Таблица ключей, указанная при помощи параметра `--keytab`, используется только для аутентификации при инициации регистрации на сервере. Во время повторной регистрации СИА создает новую таблицу для клиента.

19.4.6. Переименование клиентских машин

В данном разделе объясняется, как переименовать клиента СИА. Это действие выполняется с помощью следующих шагов:

1) Идентификация выполняемой службы и конфигурации таблицы ключей.
2) Удаление клиентской машины из домена СИА.
3) Повторная регистрация клиента на сервере с новым именем хоста.

Примечание. Переименование клиента выполняется вручную. Этот процесс рекомендуется только в крайних случаях, когда смена имени хоста абсолютно необходима.

19.4.6.1. Идентификация выполняемой службы и конфигурации таблицы ключей

Перед удалением клиента зафиксируйте некоторые его параметры. Эту конфигурацию нужно будет применить после повторной регистрации машины с новым именем хоста.

Идентификация выполняемой на машине службы:

1) Запустите команду ipa service-find и определите службы с сертификатами в ее вы-

воде:

```
$ ipa service-find client.test.dom
```

Кроме того, у каждого хоста есть служба по умолчанию, которая отсутствует в выводе ipa service-find. Принципал службы хоста, также называемый принципалом хоста, это host/client.test.dom.

2) Идентифицируйте все группы хоста, к которым принадлежит машина

```
# ipa hostgroup-find client.test.dom
```

3) Для всех принципалов служб, показанных в выводе ipa service-find client.test.dom, определите местоположение соответствующих таблиц ключей на client.test.dom.

Каждая служба в клиентской системе имеет принципал Kerberos в виде service_name/hostname@<область>, например, ldap/client.test.dom@test.dom.

19.4.6.2. Удаление клиентской машины из домена СИА

Удалите регистрацию клиентской машины из домена СИАв соответствии с инструкцией, приведенной в п. 19.4.4.3. «Удаление клиента». Удалите старые принципалы для каждой определенной таблицы ключей, кроме /etc/krb5.keytab:

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r test.dom
```

На сервере СИА удалите запись хоста. Это действие удаляет все службы и отзывает все сертификаты, выданные для этого хоста:

```
[root@server ~]# ipa host-del client.test.dom
```

Хост становится полностью удаленным из СИА.

19.4.6.3. Повторная регистрация клиента на сервере с новым именем хоста

Переименуйте машину согласно требованиям.

Повторно выполните регистрацию в качестве клиента СИА. См. подраздел «Повторная регистрация клиента в домене СИА». На сервере добавьте новую таблицу ключей для каждой службы, идентифицированной согласно параграфу 19.4.6.1. «Идентификация выполняемой службы и конфигурации таблицы ключей».

```
[root@server ~]# ipa service-add service_name/new_host_name
```

Создайте сертификаты для служб, которым были выданы сертификаты, согласно подразделу «Идентификация выполняемой службы и конфигурации таблицы ключей». Это можно сделать:

1) С помощью административных утилит СИА.
2) С помощью утилиты *certmonger* (certmonger(8)).

Повторно добавьте клиента в группы хоста согласно параграфу 19.4.6.1. «Идентификация выполняемой службы и конфигурации таблицы ключей».

Лист регистрации изменений									
Изм.	*Номера листов (страниц)*				*Всего листов (страниц) в докум.*	*№ документа*	*Входящий № сопроводительного докум. и дата*	*Подп.*	*Дата*
	измененных	*замененных*	*новых*	*аннулированных*					

Ф.19.604-1