УТВЕРЖДЕН

МСВСфера 9 Сервер

Руководство администратора

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

Дата последнего изменения: 1 ноября 2025 г.

Оглавление

Аннотация **5**

Общие сведения **6**
Назначение и область применения . . . 6
Обеспечение безопасности и требования к администратору . . . 6

Установка и начальная настройка системы **9**
Системные требования . . . 9
Создание загрузочного USB-носителя и запись iso-образа дистрибутива . . . 9
Установка системы с USB-носителя . . . 14

Графический интерфейс средств настройки системы **20**
Настройки даты и времени . . . 20
Управление пользователями . . . 23
Просмотр системных журналов . . . 31
Создание защищённых каналов связи (VPN) . . . 35
Ограничение времени работы за компьютером . . . 46
Подключение к удалённому рабочему столу . . . 57
Централизованная аутентификация и авторизация пользователей . . . 62

Настройка оборудования **82**
Управление принтерами . . . 82
Удалённое подключение USB-устройств по сети . . . 86

Управление пакетами **92**
Введение и основные понятия . . . 92
Пакетный менеджер DNF . . . 93
Безопасность . . . 99
Графический интерфейс к менеджеру пакетов DNF dnfdragora . . . 99

Идентификация и аутентификация **104**
Введение . . . 104
Добавление нового пользователя . . . 104
Изменение уже имеющихся пользовательских записей . . . 105
Удаление пользователей . . . 106
Добавление группы пользователей . . . 106
Изменение существующей группы пользователей . . . 107
Удаление существующей группы пользователей . . . 107
Создание и изменение пароля пользователя . . . 107
Изменение срока действия учётной записи и пароля пользователя . . . 109
Управление политиками паролей . . . 109
Получение сведений о пользователе . . . 116
Конфигурационный файл /etc/login.defs . . . 117
Конфигурационный файл /etc/pam.d/system-auth . . . 118
Конфигурационный файл /etc/issue . . . 121
Конфигурационный файл /etc/shadow . . . 122

Запуск программ от имени другого пользователя . . . 122

Управление доступом **134**
Введение . . . 134
Установка и изменение прав доступа к файлам и директориям . . . 134
Назначение и изменение владельца файла и директории . . . 135
Изменение группы-владельца файла или директории . . . 135
Просмотр и изменение списков правил контроля доступа для файлов и директорий . . . 136
Просмотр списков контроля доступа . . . 136
Редактирование пользовательских квот для файловой системы . . . 138
Конфигурационный файл /etc/profile . . . 139
Конфигурационный файл /etc/security/limits.conf . . . 141
Конфигурационный файл /etc/fstab . . . 144

Регистрация событий безопасности **146**
Введение . . . 146
Настройка сервиса auditd . . . 148
Управление правилами аудита . . . 154
Работа с журналом событий безопасности . . . 167

Ограничение программной среды **194**
Введение . . . 194
Включение программ в автозагрузку . . . 194
Управление системными службами . . . 194
Настройка запуска программ по расписанию . . . 195
Управление программными пакетами . . . 196
Установка последней версии пакета/группы пакетов . . . 197

Стирание данных **198**
Введение . . . 198
Заполнение случайными числами места, занятого файлами . . . 198
Стирание данных в свободном пространстве раздела, в котором находится директория . . . 198
Стирание данных в разделах подкачки . . . 199
Стирание данных в оперативной памяти . . . 199

Контроль целостности **201**
Введение . . . 201
Вычисление и сверка контрольной суммы файла . . . 201
Проверка целостности данных . . . 202

Обеспечение надёжного функционирования **204**
Введение . . . 204
Архивация файлов и директорий . . . 204
Создание архивов и извлечение файлов из них . . . 205

Резервное копирование данных . . . 205
Создание дисковых RAID-массивов . . . 206

Фильтрация сетевого потока **208**
Введение . . . 208
Настройка файрвола (брандмауэра) . . . 208
Конфигурационный файл /etc/firewalld/firewalld.conf . . . 209

Создание виртуальной машины **211**
Создание виртуальной машины с помощью утилиты virt-install . . . 211

Контейнеризация **215**
Trivy . . . 215

Панель управления Cockpit **219**
Описание панели управления Cockpit . . . 219
Установка и настройка Cockpit . . . 220
Создание диагностических отчётов . . . 220
Настройка мультитерминального режима . . . 224
Расширение USBGuard для Cockpit . . . 229
Расширение Bootloader для Cockpit . . . 236
Подключение к домену . . . 240
Расширение Aide для Cockpit . . . 244
Расширение LibreOffice для Cockpit . . . 248
Расширение Quota для Cockpit . . . 250
Расширение «Виртуальные терминалы» для Cockpit . . . 255

Аннотация

Настоящее руководство ориентировано на специалистов, знакомых с операционными системами типа Linux и имеющих минимальный практический опыт работы с ними.

Руководство предназначено для администраторов серверной операционной системы с интегрированными серверными службами МСВСфера 9 Сервер.

Руководство снабжено примерами, сделанными в операционной системе МСВСфера 9 Сервер, установленной в базовой конфигурации.

Общие сведения

Назначение и область применения

МСВСфера Сервер — серверная операционная система на основе ядра Linux с набором интегрированных служб и приложений, включающим веб-сервер, почтовый сервер, сервер служб сетевой инфраструктуры, серверы файлов и печати, средства резервного копирования и восстановления данных, множество других служб и приложений, а также средства администрирования и защиты информации. Развёрнутую ОС МСВСфера Сервер применяют в качестве программной платформы для использования в корпоративных сетях и серверных окружениях.

Обеспечение безопасности и требования к администратору

Внедрению и использованию операционной системы должны предшествовать подготовительные процедуры, направленные на обеспечение безопасности при приемке установочного дистрибутива операционной системы от поставщика, на обеспечение безопасной установки, настройки и запуска операционной системы и на создание безопасной среды её функционирования. Реализация подготовительных процедур должна обеспечиваться необходимыми ресурсами и сопровождаться назначением ответственных за их выполнение должностных лиц.

Процедуры безопасной приемки должны предусматривать меры подтверждения подлинности установочного дистрибутива операционной системы, исключающие возможности преднамеренного или непреднамеренного внесения изменений в поставляемую версию, т.е. замены её фальсифицированной или неработоспособной версией.

К таким мерам в общем случае относятся:

- проверка подлинности источника поставки путем визуального контроля наличия и целостности специальных защитных стикеров (наклеек, знаков) на упаковке комплекта поставки, а также целостности самой упаковки;
- проверка комплектности поставки в соответствии с заявкой, договорными материалами и спецификацией, сверка маркировки и номера версии;
- проверка целостности установочного дистрибутива с помощью программного средства контроля целостности путем сравнения с эталонным значением контрольной суммы или с помощью средств электронной подписи.

Процедуры безопасной установки, настройки, запуска операционной системы и создания безопасной среды её функционирования в общем случае должны предусматривать меры, обеспечивающие:

- совместимость операционной системы со средствами вычислительной техники, на которых планируется её установка и использование;
- установку, конфигурирование, настройку, запуск и управление операционной системой в соответствии с эксплуатационной документацией и принятой

политикой безопасности;

- защиту от действий, направленных на нарушение физической целостности средств вычислительной техники, на которых она функционирует;
- доверенную загрузку операционной системы, контроль доступа к процессу загрузки, блокирование попыток несанкционированной загрузки, контроль целостности компонентов загружаемой операционной среды;
- наличие ресурсов для выполнения функциональных возможностей безопасности операционной системы, хранения создаваемых резервных копий, а также защищенное хранение данных операционной системы и защищаемой информации;
- ограничение на установку программного обеспечения и его компонентов, не задействованных в технологическом процессе обработки информации;
- доверенный маршрут между операционной системой и пользователями;
- доверенный канал передачи данных между операционной системой и средствами вычислительной техники, на которых происходит обработка информации, а также с которых происходит их администрирование;
- невозможность отключения или обхода компонентов операционной системы и средств защиты информации;
- препятствие несанкционированному копированию информации, содержащейся в операционной системе, на съемные носители информации, в том числе контроль вноса (выноса) в (из) контролируемую зону съемных носителей информации;
- проверку целостности получаемых от поставщика внешних модулей уровня ядра перед их установкой в операционную систему;
- выделение вычислительных ресурсов для процессов в соответствии с их приоритетами;
- профессиональную компетентность и надежность персонала, ответственного за администрирование системы, его способность выполнять свои обязанности в точном соответствии с принятой политикой безопасности и эксплуатационной документацией;
- возможность генерации аутентификационной информации, соответствующей заданной метрике качества;
- недоступность аутентификационной информации для лиц, не уполномоченных на её использование;
- разделение полномочий пользователей и администраторов с назначением им минимально необходимых прав и привилегий;
- исключение в процессе использования системы доступа пользователей к приложениям, выполняющимся с более высокими правами доступа, чем права,

предоставленные им согласно матрице доступа;

- завершение администраторами приложений, запущенных ими с административными правами, после окончания работы с ними;
- запрет пользователям на передачу посторонним лицам своей личной идентификационной и аутентификационной информации, а также на регистрацию кого-либо в системе под своим именем и паролем.

Для управления ОС МСВСфера используются командные интерпретаторы (shell). Поэтому администратор должен иметь:

- базовые навыки администрирования ОС семейства Linux;
- навыки конфигурирования и настройки программных продуктов и ОС;
- опыт работы со стандартными элементами графического интерфейса приложений;
- навыки поддержания в работоспособном состоянии технических средств ПК.

Установка и начальная настройка системы

Системные требования

Для установки операционной системы МСВСфера вам понадобится устройство со следующими характеристиками.

Минимальные

Для использования операционной системы требуется компьютер со следующими минимальными характеристиками:

- Процессор:
  - Intel или AMD версии не ниже x86-64-v2 (Intel Nehalem и более поздние, AMD Bulldozer и более поздние).
  - 64-битной архитектуры от ARM — aarch64 (ОС МСВСфера 9.5 и более поздние версии).
- 2 Гбайта оперативной памяти.
- 20 Гбайт свободного пространства памяти на жёстком диске в зависимости от используемой конфигурации.

Рекомендуемые

Для полнофункционального использования операционной системы рекомендуется использовать компьютер со следующими характеристиками:

- Процессор:
  - Intel или AMD версии не ниже x86-64-v2 (Intel Nehalem и более поздние, AMD Bulldozer и более поздние).
  - 64-битной архитектуры от ARM — aarch64 (ОС МСВСфера 9.5 и более поздние версии).
- 8 Гбайт оперативной памяти.
- 40 Гбайт свободного пространства памяти на жёстком диске в зависимости от используемой конфигурации.

Создание загрузочного USB-носителя и запись iso-образа дистрибутива

В настоящее время наиболее удобным способом установки операционной системы МСВСфера является использование USB-носителя с записанным на него дистрибутивом. Ниже мы рассмотрим, как создать загрузочный USB-носитель и записать на него iso-образ дистрибутива.

Программное обеспечение, рекомендуемое для создания загрузочного USB-носителя и записи iso-образа дистрибутива МСВСфера:

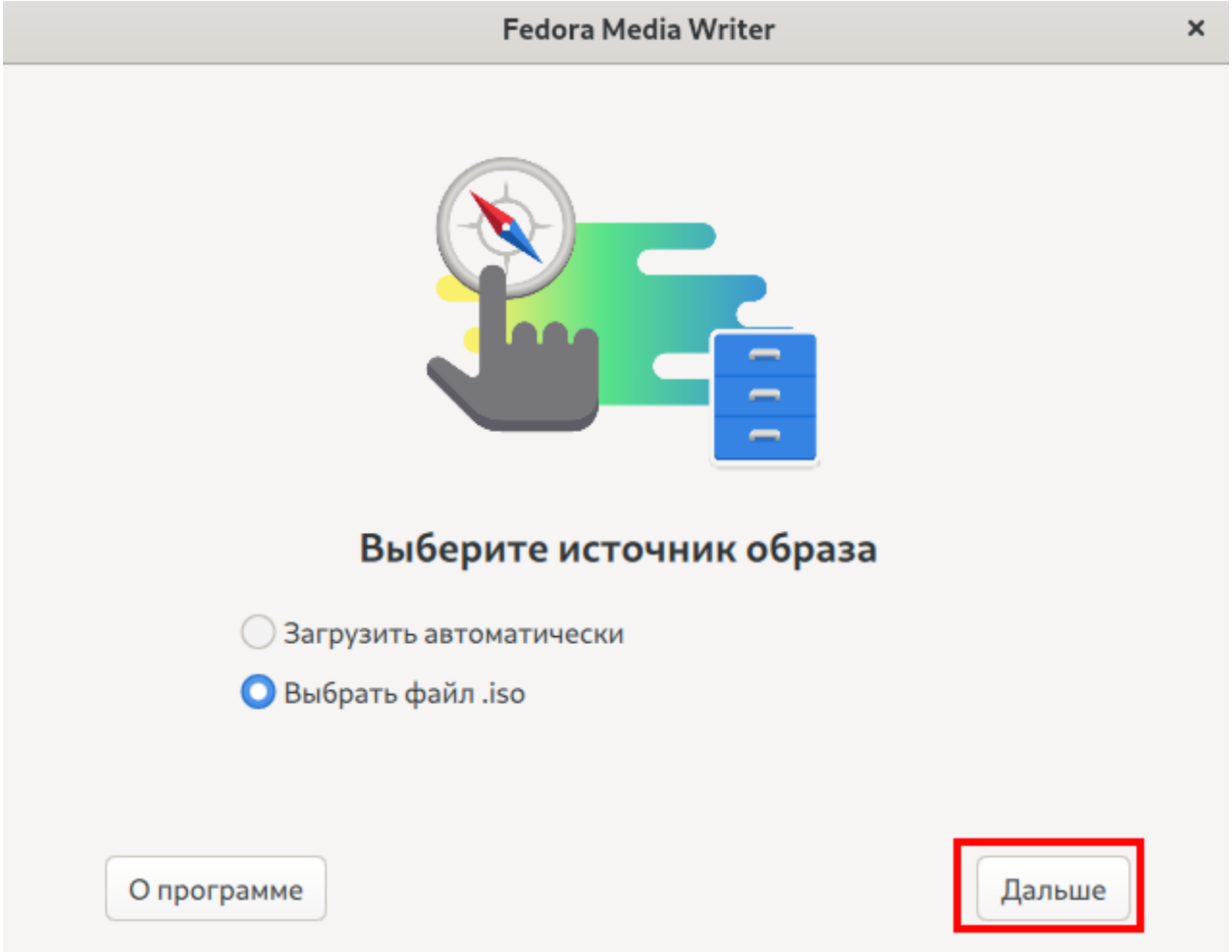
- Fedora Media Writer — для операционных систем семейства Windows, Linux и macOS;
- balenaEtcher — для операционных систем семейства Windows, Linux и macOS;
- Win32 Disk Imager — для операционных систем семейства Windows;
- Утилита командной строки dd — для операционных систем семейства Linux.

Интерфейс указанного программного обеспечения интуитивно понятный, дополнительные инструкции вы можете найти в документации соответствующего ПО.

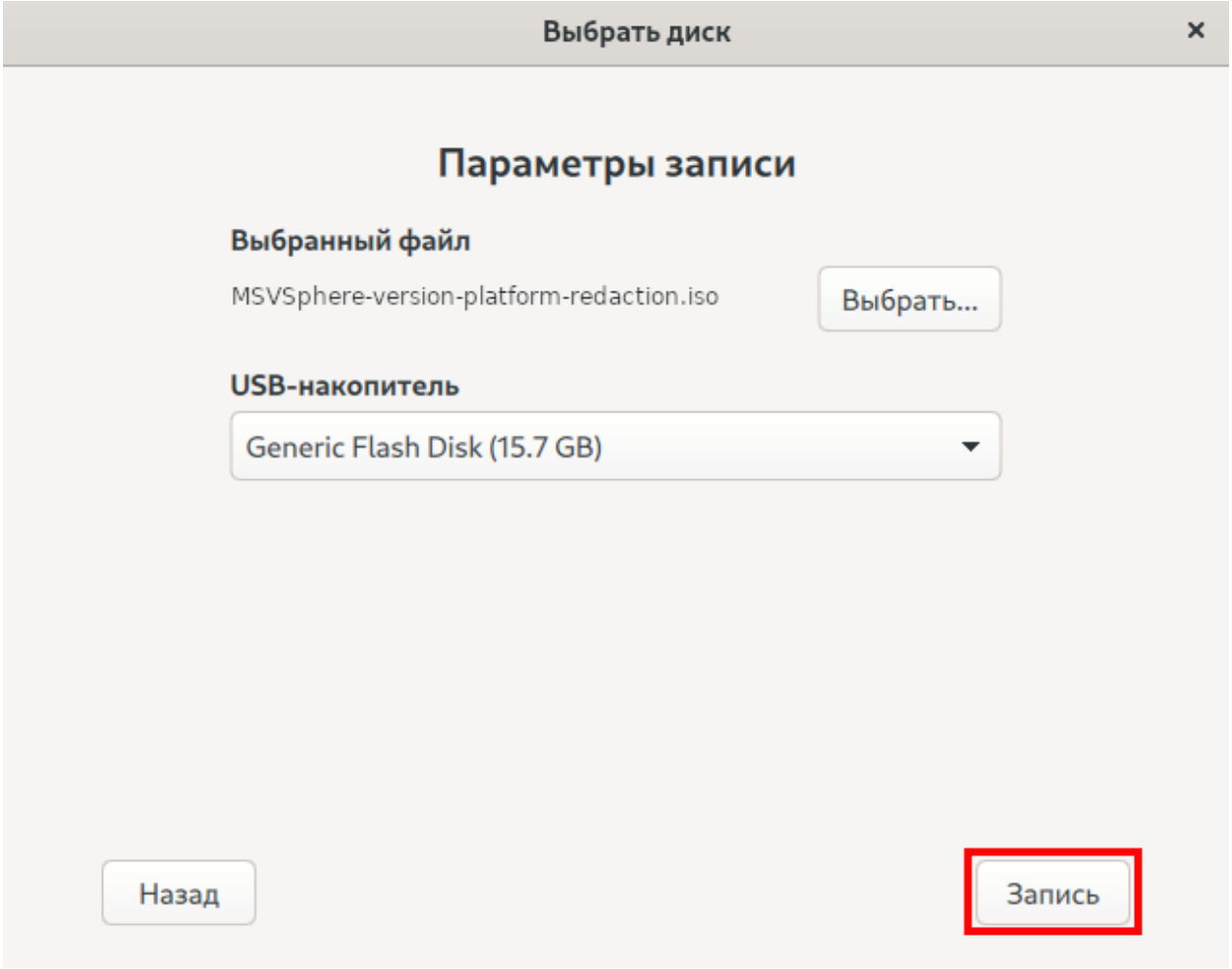
В качестве примера рассмотрим процесс создания загрузочного USB-носителя и записи iso-образа дистрибутива ОС МСВСфера в программе Fedora Media Writer в операционной системе семейства Windows и с использованием утилиты командной строки `dd` в операционной системе семейства Linux.

Пример создания загрузочного USB-носителя и записи iso-образа дистрибутива ОС МСВСфера в программе Fedora Media Writer (Windows)

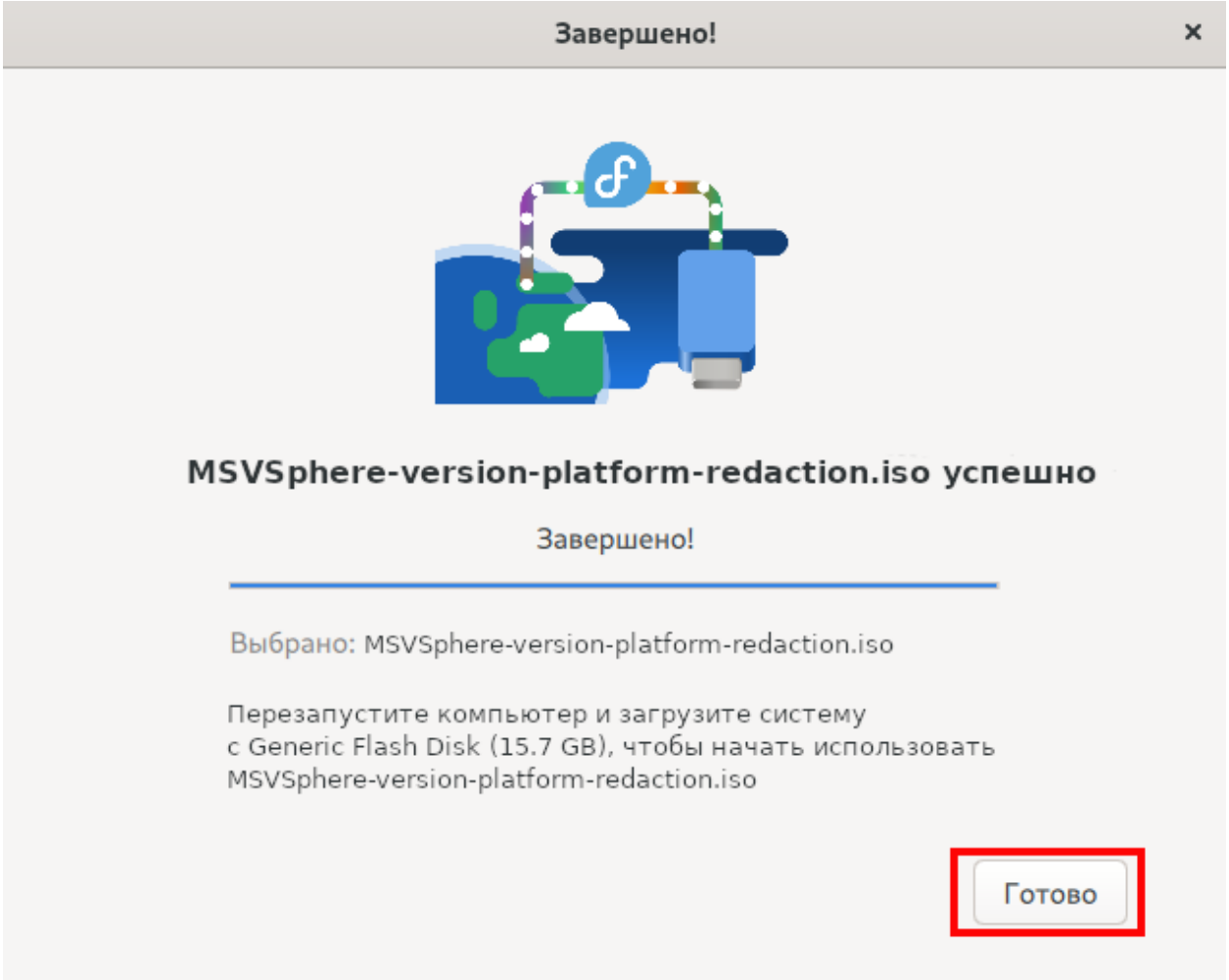
1. Скачайте последнюю версию Fedora Media Writer для Windows на ваше устройство.
2. Запустите установочный файл и выполните установку Fedora Media Writer на ваше устройство.
3. Вставьте USB-носитель, на который вы планируете записывать iso-образ дистрибутива. Убедитесь, что на нём достаточно места.
4. Скачайте актуальный iso-образ МСВСфера: https://msvsphere-os.ru/downloads/.
5. Запустите Fedora Media Writer.
6. Выберите источник образа — «Выбрать файл iso» и нажмите «Дальше».



7. В окне «Выбрать диск» → «Параметры записи» → «Выбранный файл» нажмите на кнопку «Выбрать» для выбора iso-образа ОС МСВСфера, загруженного ранее.
8. USB-накопители определяются автоматически. Если у вас подключено несколько USB-носителей, выберите необходимый из списка.
9. После выбора iso-образа ОС МСВСфера нажмите «Запись».



10. При необходимости укажите пароль администратора для подтверждения записи.
11. Начнётся запись iso-образа ОС МСВСфера на USB-носитель. Это может занять некоторое время.
12. После завершения записи нажмите «Готово».



13. Вы успешно создали загрузочный USB-носитель ОС МСВСфера! Теперь можно приступать к установке системы (см. «*Установка системы с USB-носителя*»).

Пример создания загрузочного USB-носителя и записи iso-образа дистрибутива ОС МСВСфера с помощью утилиты командной строки dd (Linux)

1. Вставьте USB-носитель, на который вы планируете записывать iso-образ дистрибутива. Убедитесь, что на нём достаточно места.
2. Скачайте актуальный iso-образ ОС МСВСфера: https://msvsphere-os.ru/downloads/.
3. Откройте «Терминал».
4. Введите команду для записи iso-образа:

```
$ sudo dd oflag=dsync if=MSVSphere-version-platform-redaction.iso of=/dev/sdc bs=1M␣
↪status=progress;sync
```

Где `version-platform-redaction` — актуальные данные для системы, которую вы устанавливаете.

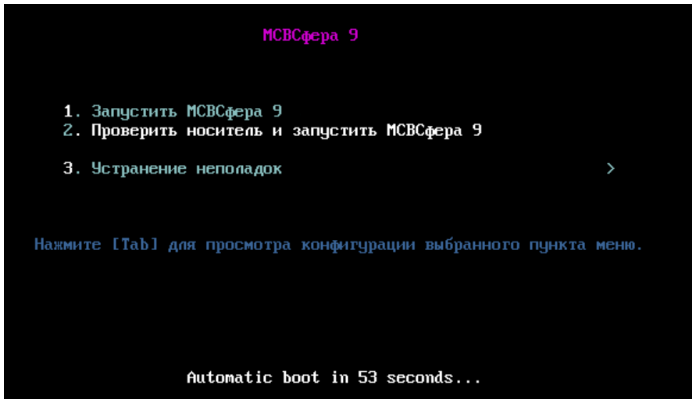
Установка системы с USB-носителя

Для установки ОС МСВСфера с USB-носителя необходимо перед началом установки выбрать приоритетную загрузку с USB-носителя в BIOS устройства, либо выбрать загрузку с USB-носителя однократно в процессе инициализации компьютера.

Для установки и загрузки ОС МСВСфера может потребоваться отключить параметр Secure Boot в BIOS устройства, на которое производится установка.

Для начала установки подключите USB-носитель с установочным дистрибутивом к компьютеру.

Сначала установка будет проходить в текстовом режиме.



Доступны следующие варианты:

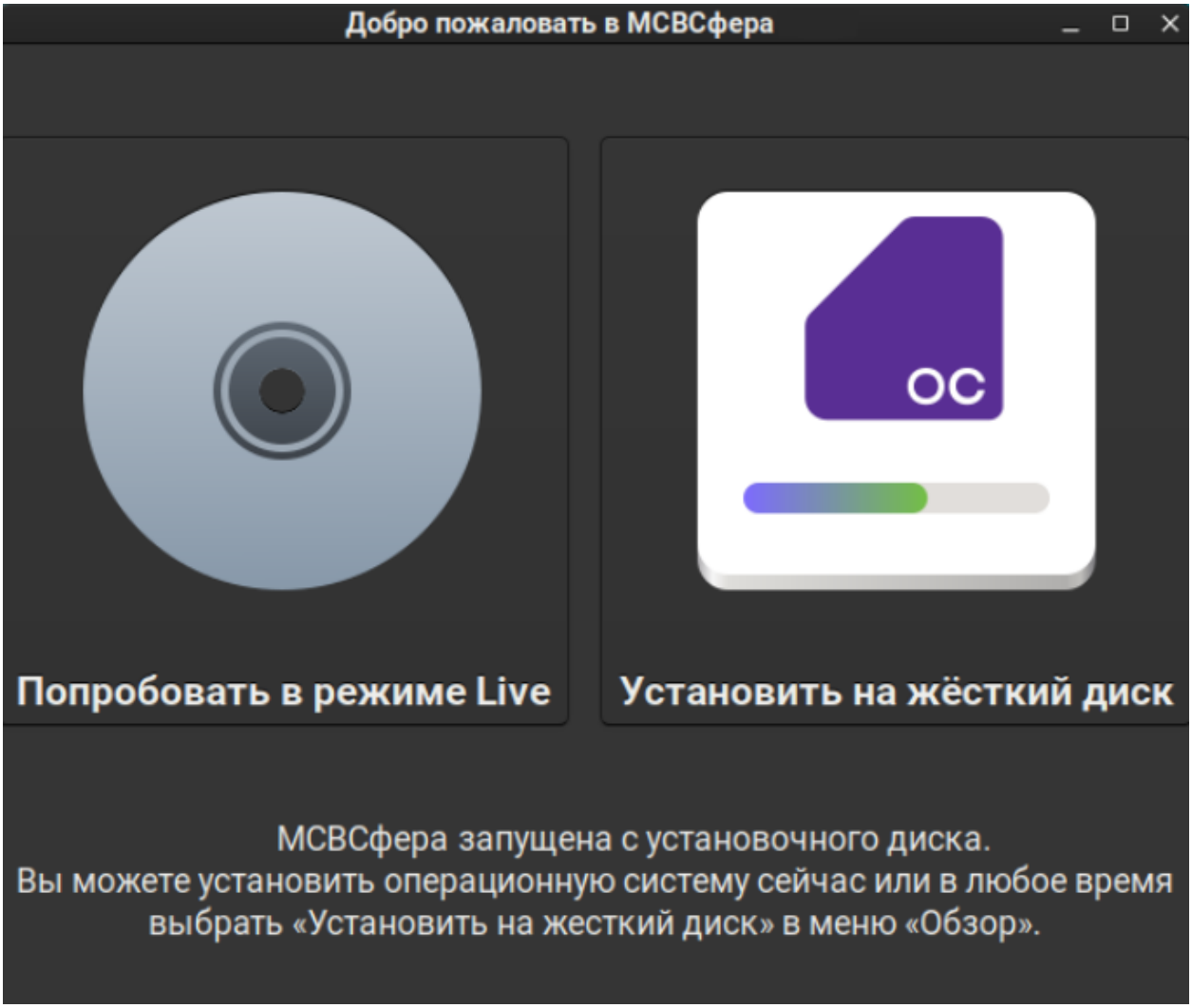
- Запустить МСВСфера 9 — начнётся установка МСВСфера 9 на ваше устройство.
- Проверить носитель и запустить МСВСфера 9 — программа установки проверит контрольные суммы образа диска, подтверждая что скачивание образа и запись на загрузочный носитель прошли без ошибок.
- Устранение неполадок — вы сможете перейти в режим восстановления, который представляет собой минимальную среду МСВСфера 9, загружаемую с загрузочного носителя. В этом режиме используются утилиты командной строки, с помощью которых вы можете монтировать или не монтировать файловые системы, заносить в чёрный список и добавлять драйверы, устанавливать и обновлять системные пакеты, а также управлять разделами.

При нажатии на «Запустить МСВСфера 9 система будет запущена с установочного диска и готова для работы в режиме Live. В этом режиме вы можете ознакомиться с функциональными возможностями МСВСфера 9 без установки системы на жёсткий диск, а также проверить совместимость и корректную работу программного и аппаратного обеспечения.

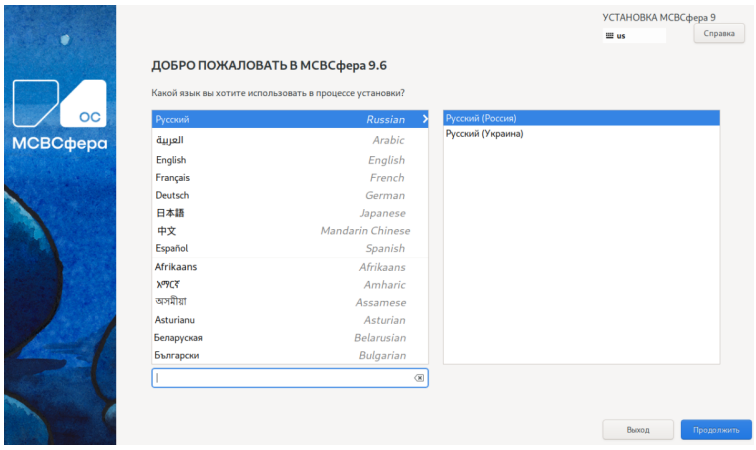
Примечание

Обратите внимание, что все настройки, выполненные в режиме Live, будут потеряны (не сохранятся) после перезагрузки.

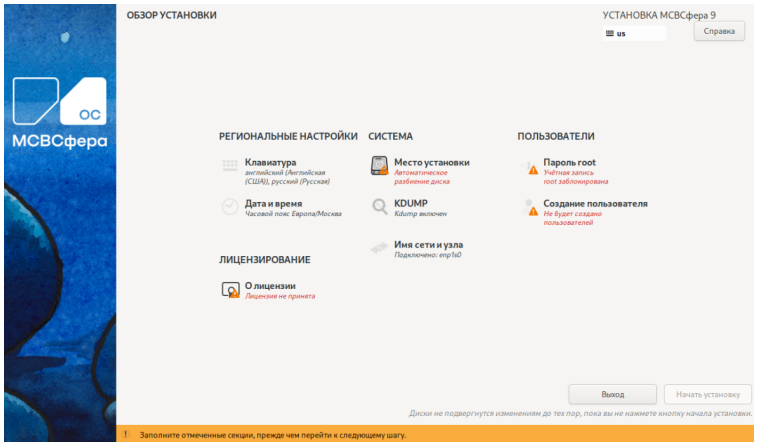
Для полноценной установки МСВСфера 9 выберите «Установить на жёсткий диск».



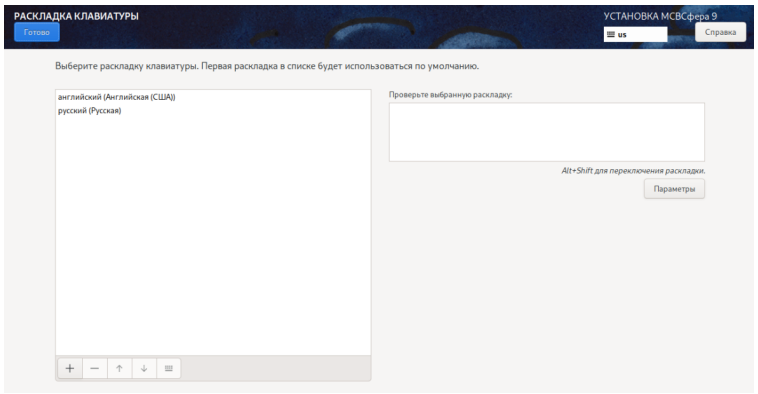
После этого установка продолжится в графическом режиме и на экране монитора компьютера появится окно с предложением выбрать язык установки.



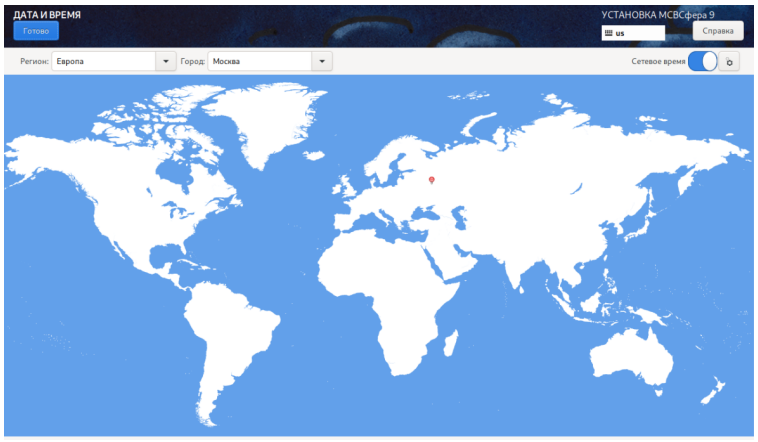
Затем появится окно «Обзор установки», с помощью которого, последовательно нажимая кнопку «Готово», можно будет произвести все необходимые настройки.



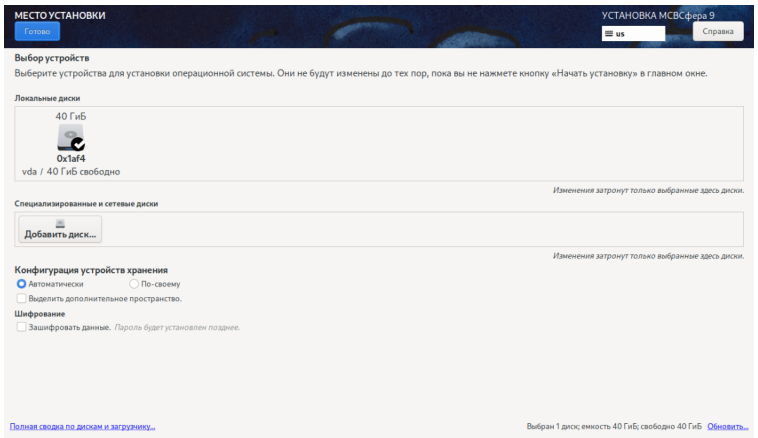
Раскладка клавиатуры.



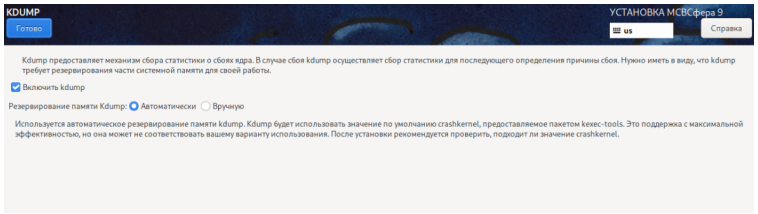
Дата и время.



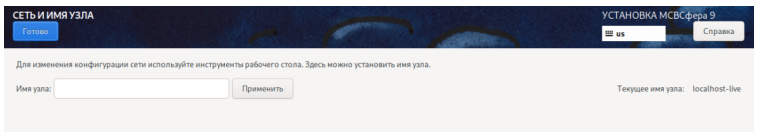
Место установки.



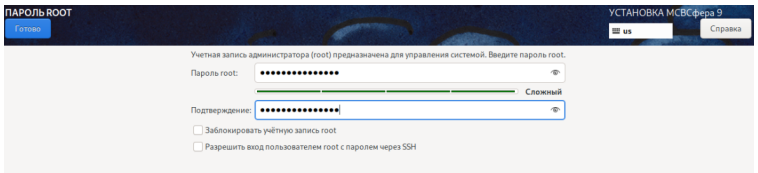
Диагностика сбоев ядра.



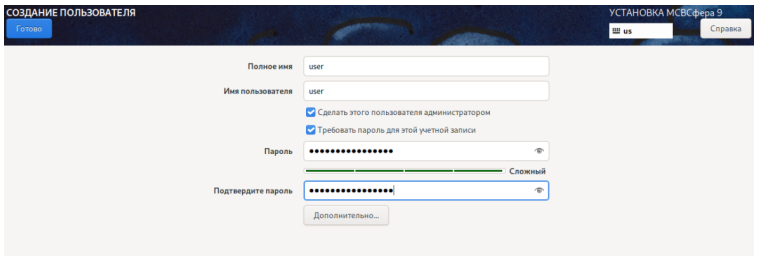
Имя сети и узла.



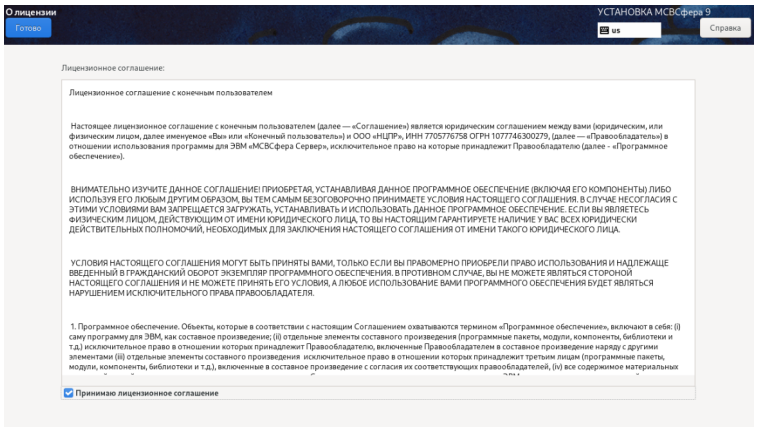
При необходимости создайте суперпользователя root.



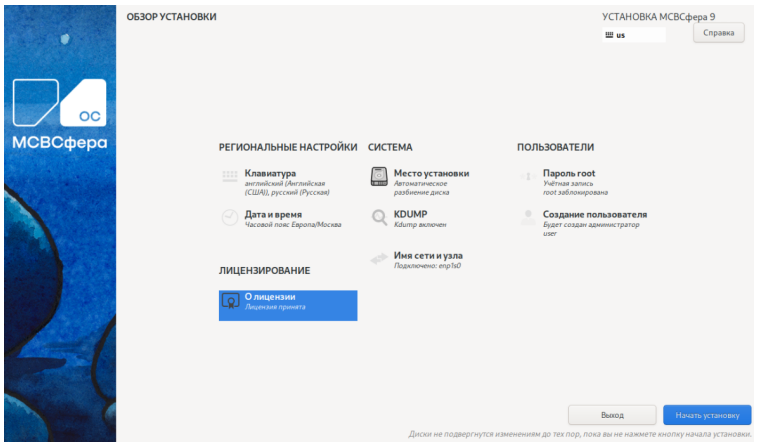
Обязательно создайте как минимум одного пользователя.



Перейдите в раздел «О лицензии», прочитайте и примите лицензионное соглашение.

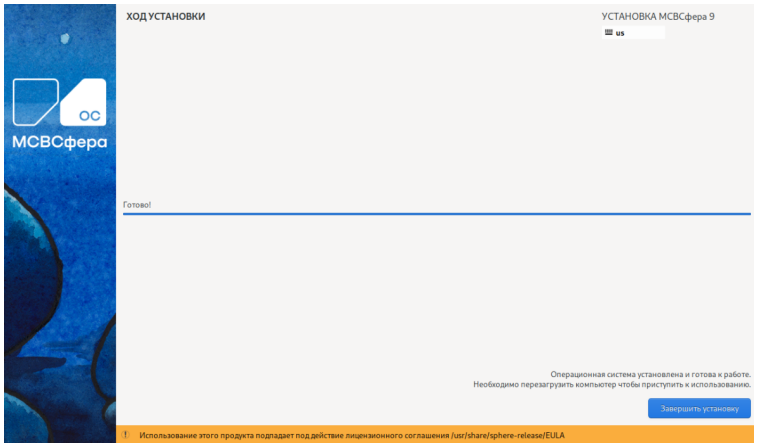


После того, как все необходимые настройки произведены, нажмите на кнопку «Начать установку» и процесс установки начнётся.



Продолжительность установки может составить примерно 20-30 минут, в зависимости от быстродействия оборудования и выбранной конфигурации программного обеспечения.

По завершении установки на экране монитора появится соответствующее уведомление с предложением произвести перезагрузку.



После извлечения USB-носителя с установочным дистрибутивом и перезагрузки системы появится приглашение войти в систему, пройдя идентификацию и аутентификацию.

Для того, чтобы появилась возможность подключать разделы с файловой системой BTRFS необходимо после установки системы установить пакет `kmod-btrfs`:

```
$ sudo dnf install kmod-btrfs
```

Графический интерфейс средств настройки системы

Настройки даты и времени

Введение

Настройка даты и времени в графическом интерфейсе ОС МСВСфера производится в приложении «**Настройки**». Перейти в «**Настройки**» вы можете из главного меню, набрав в строке поиска «настройки» и нажав на приложение правой кнопкой мыши, или нажав значок «**Шестерёнки**» в системной панели или в главном меню.

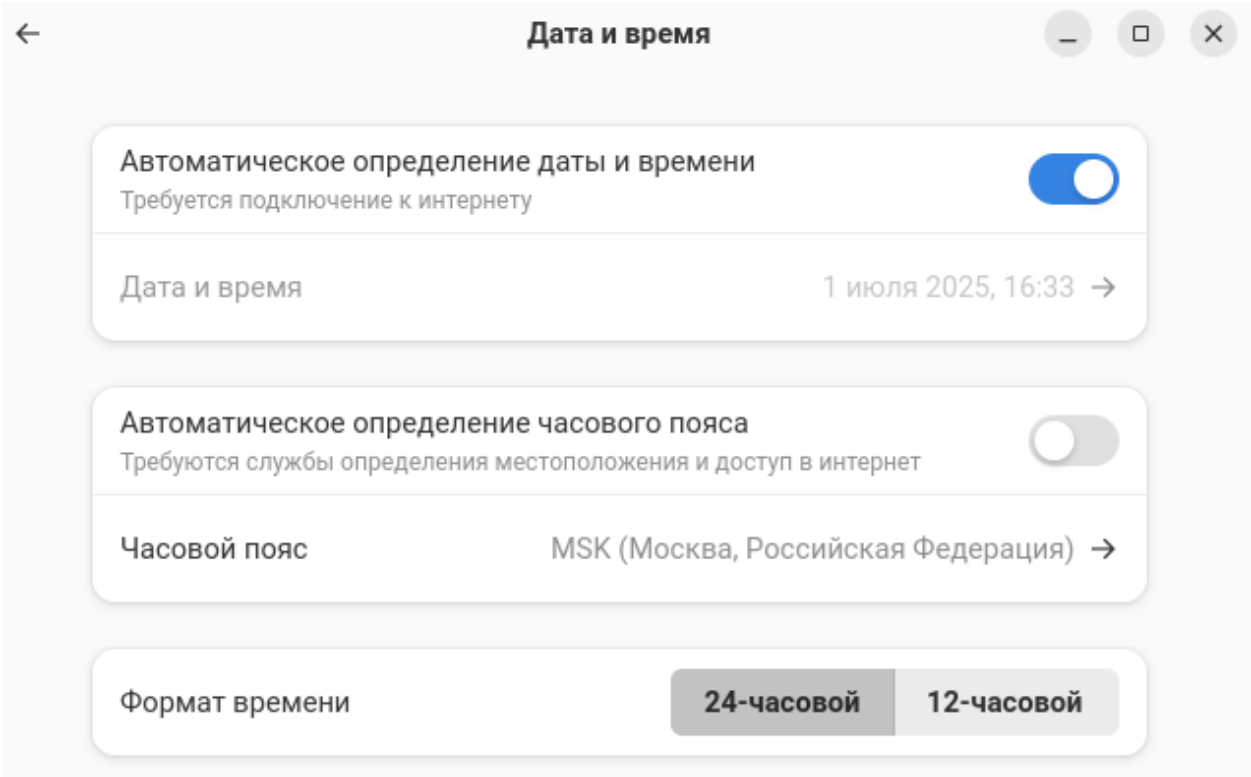
Зачастую настройка даты и времени производится при начальной установке системы, но вы всегда можете изменить изначально заданную конфигурацию в приложении «**Настройки**».

При изменении настроек даты и времени перезагрузка операционной системы как правило не требуется, все изменения применяются сразу.

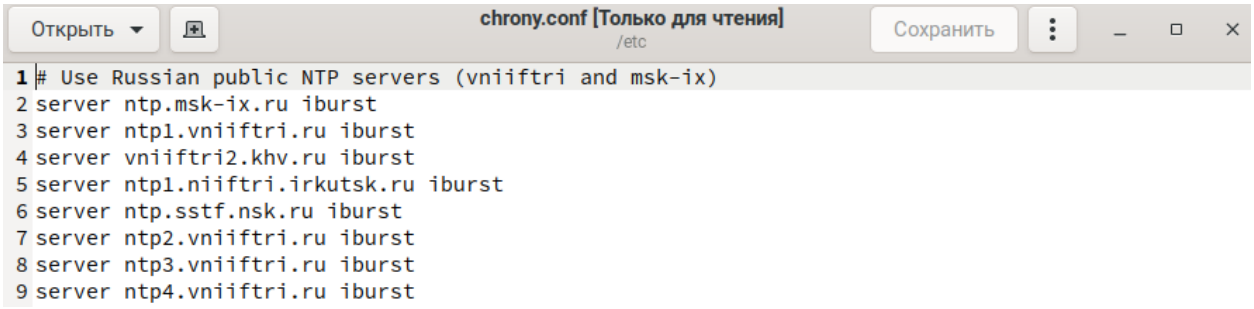
Включение и выключение автоматической синхронизации времени

Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Дата и время**».

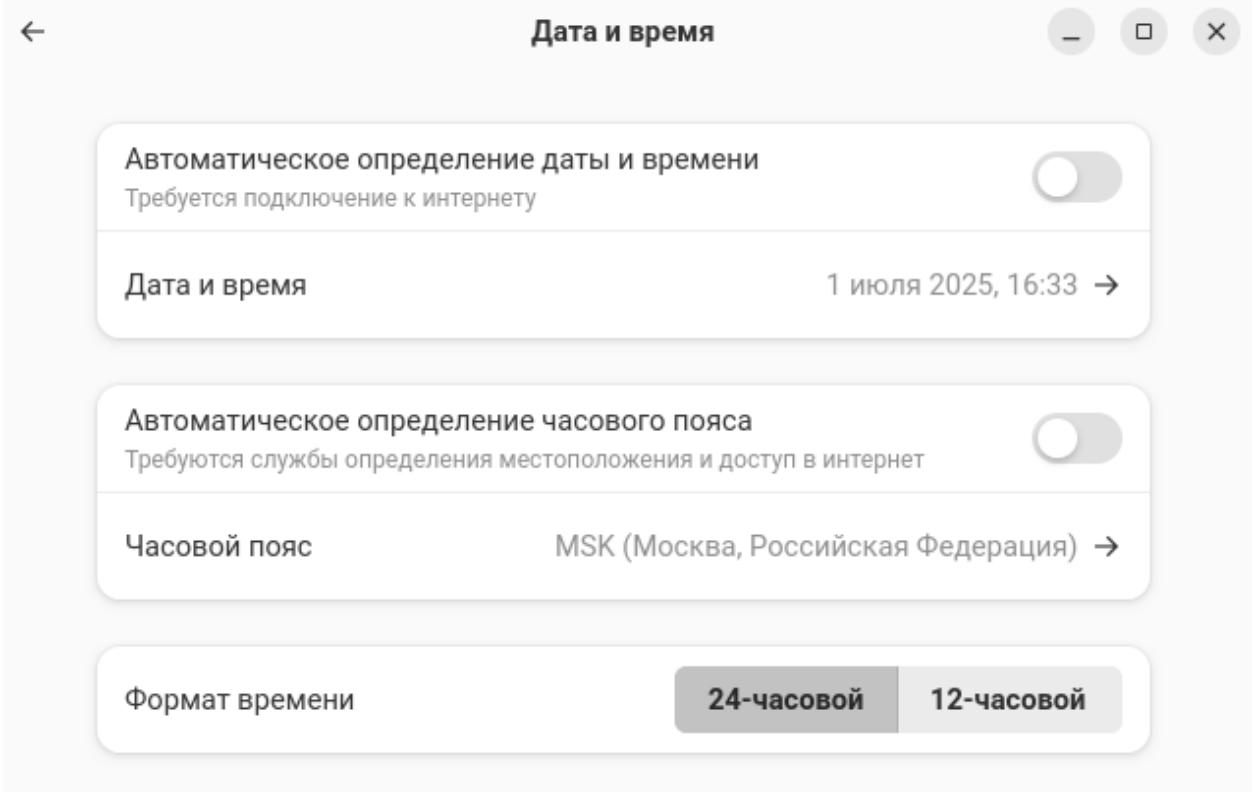
По умолчанию включено автоматическое определение даты и времени.



Для автоматической синхронизации времени используются российские NTP-серверы, список которых вы можете посмотреть в файле `/etc/chrony.conf`.



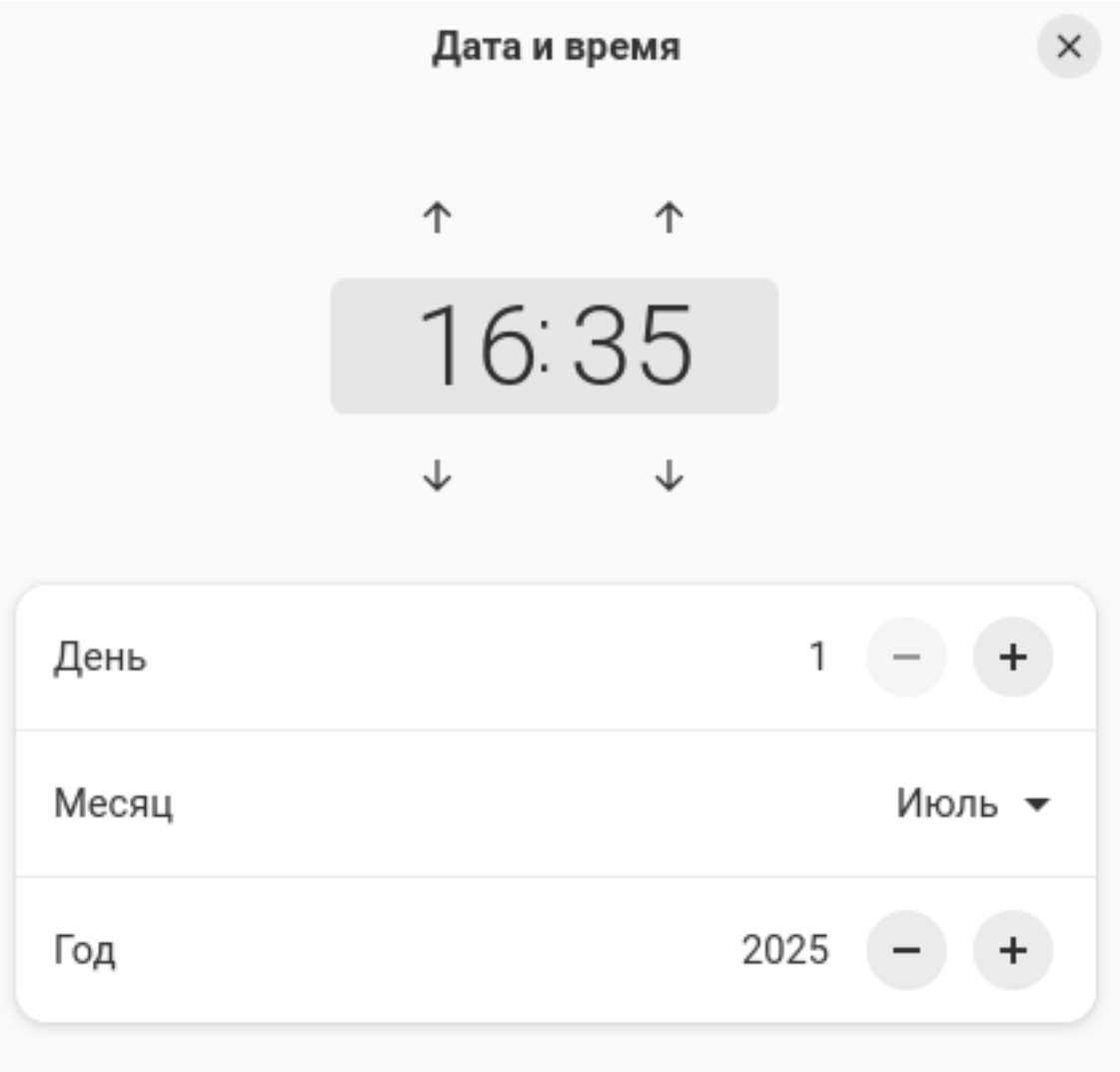
Для отключения автоматического определения даты и времени передвиньте ползунок в неактивное состояние. Вы увидите, что станет активной строка «**Дата и время**»



Установка даты и времени вручную

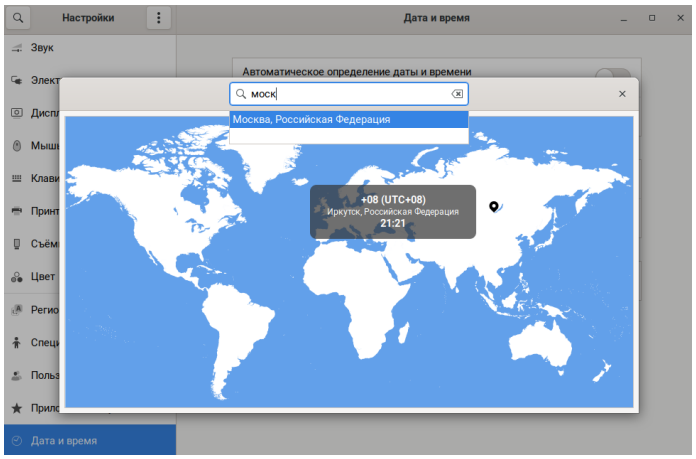
Для установки даты и времени вручную сначала отключите автоматическое определение даты и времени (см. «*Включение и выключение автоматической синхронизации времени*»).

Затем нажмите на строку «**Дата и время**», откроется окно, в котором вы можете задать дату и время вручную.



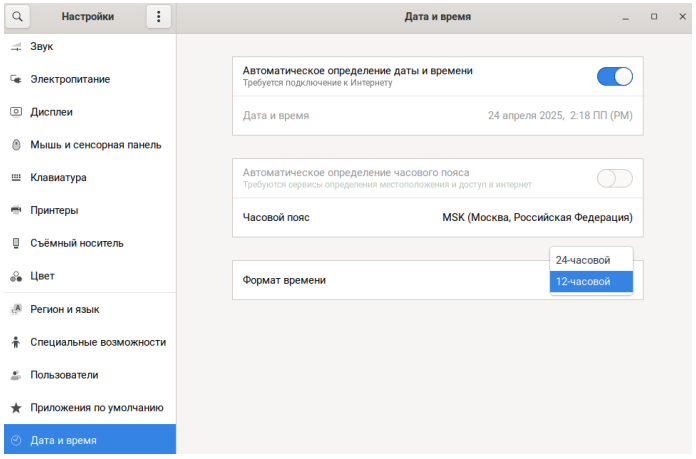
Установка и смена часового пояса

Для установки или смены часового пояса нажмите на строку «**Часовой пояс**», откроется окно, в котором отображается карта с текущим часовым поясом. Начните вводить название города/местности в строке поиска, затем выберите требуемый город из выпадающего списка. Или просто укажите точку на карте. Часовой пояс поменяется на пояс, ассоциированный с выбранным городом/местностью, что отобразится на карте. После завершения настройки закройте окно.



Изменение формата времени

Для изменения формата времени нажмите на выпадающий список в строке «**Формат времени**» и выберите требуемый вид. При 12-часовом формате времени «**ПП (PM)**» значит «после полудня», а «**ДП (AM)**» — до полудня.



Управление пользователями

Введение

Управление пользователями в графическом интерфейсе ОС МСВСфера производится в приложении «**Настройки**». Перейти в «**Настройки**» вы можете из главного меню, набрав в строке поиска «настройки» и нажав на приложение правой кнопкой мыши, или нажав значок «**Шестерёнки**» в системной панели или в главном меню.

Для пользователей, созданных при *начальной установке системы*, вы всегда можете изменить изначально заданные параметры в приложении «**Настройки**».

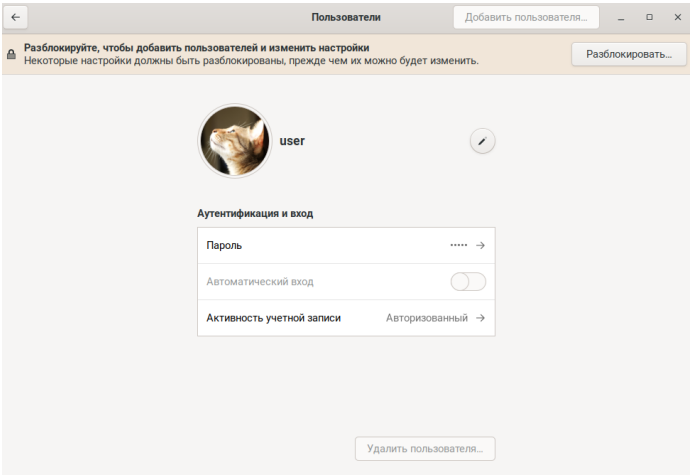
Приложение предоставляет различные функции управления пользователями, включая создание, удаление и изменение учётных записей пользователей, управление паролями, настройку прав доступа, а также администрирование групп пользователей.

Создание пользователя

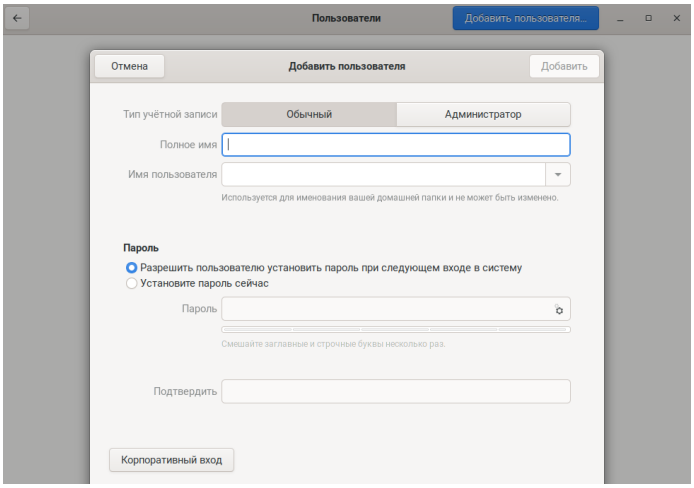
Рассмотрим процедуру создания учётной записи обычного пользователя и администратора.

Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Пользователи**».

Для любой учётной записи (кроме суперпользователя) при начальном входе некоторые настройки будут заблокированы. Для разблокировки необходимо указать пароль.



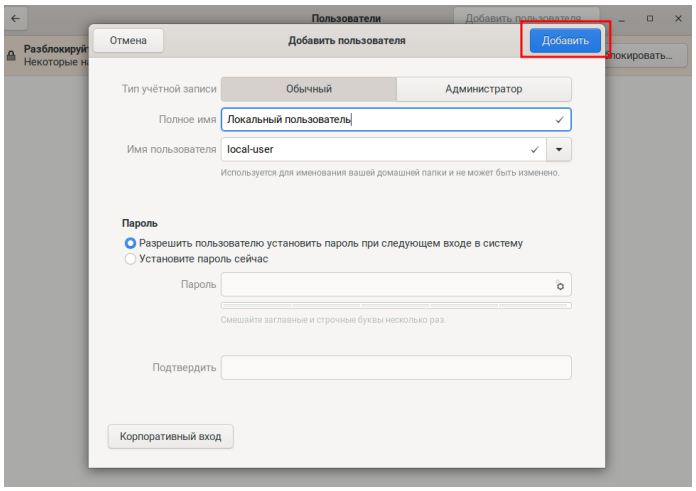
После успешного прохождения аутентификации активируется кнопка «**Добавить пользователя**» в правом верхнем углу окна. Нажмите на неё, откроется окно «**Добавить пользователя**».



Заполните необходимые поля.

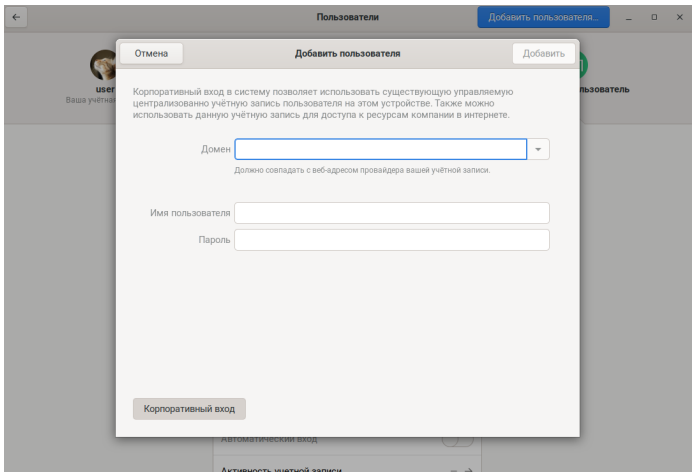
- Тип учётной записи — «Обычный» или «Администратор».
- Полное имя — обычно указывается ФИО или название роли (например, «локальный администратор»).
- Имя пользователя — обычно указывается в английской раскладке и используется для именования домашней папки пользователя. Не может быть изменено. Например, для имени пользователя `user` домашняя папка будет иметь вид `/home/user/`.
- Пароль — выберите, будет ли пароль задан при первом входе в систему или установлен при создании пользователя. При установке пароля во время создания пользователя вы можете воспользоваться функцией автоматической генерации пароля, нажав на значок «**Шестерёнки**».

После успешного заполнения всех полей активируется кнопка «**Добавить**» в правом верхнем углу. Нажмите её для добавления пользователя. По запросу системы снова выполните аутентификацию. Пользователь будет добавлен и сразу отобразится в приложении «**Настройки**» — «**Пользователи**»



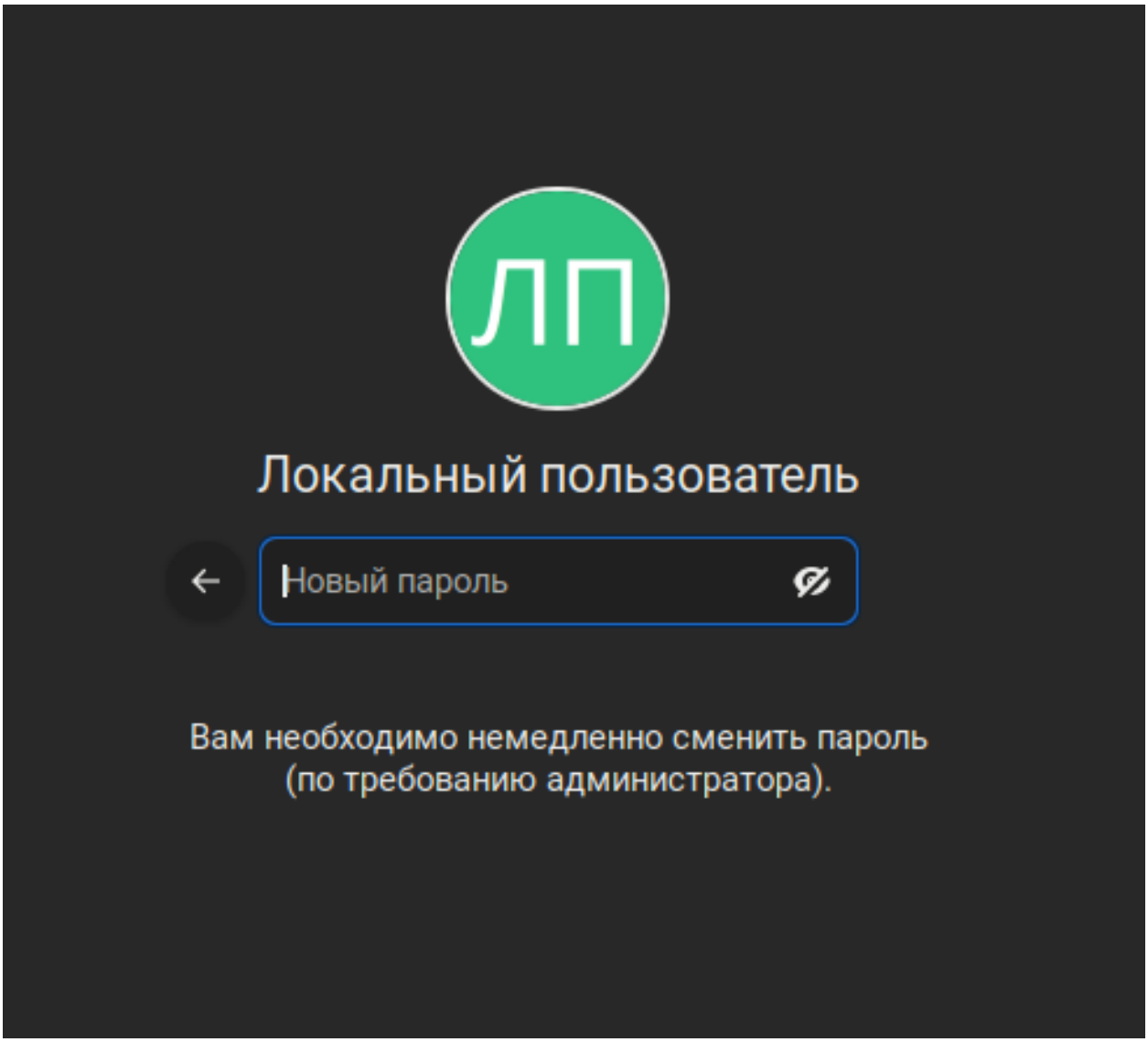
Корпоративный вход

Если в вашей компании используется централизованное управление доступом и учётными записями, то вы можете добавить корпоративную учётную запись пользователя сразу при его создании. Необходимые данные для корпоративного входа обычно могут быть предоставлены администраторами вашей компании.

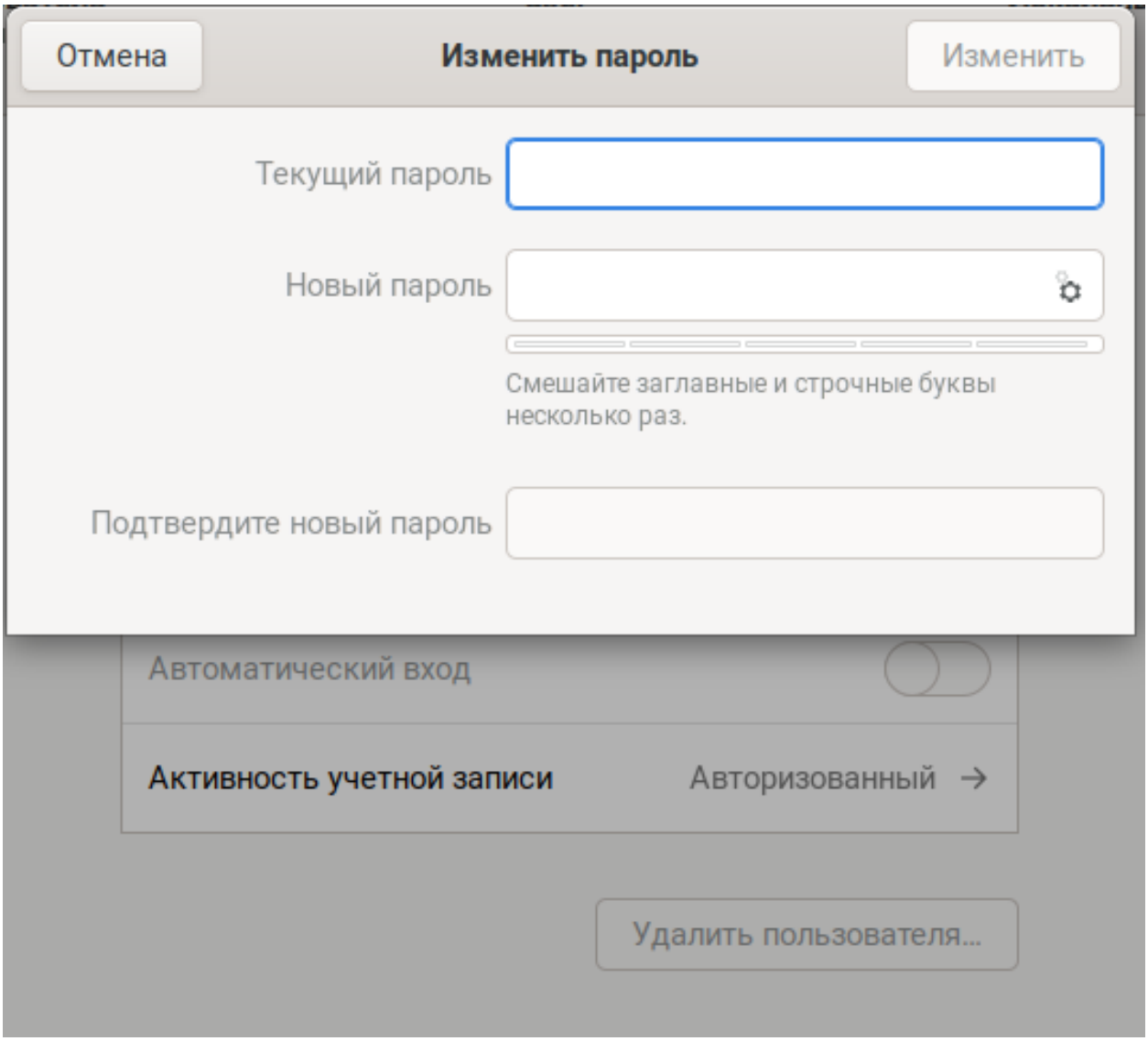


Установка и изменение пароля пользователя

Если при создании учётной записи пользователя было выбрано задание пароля при первом входе в систему, то пользователю будет предложено установить пароль.



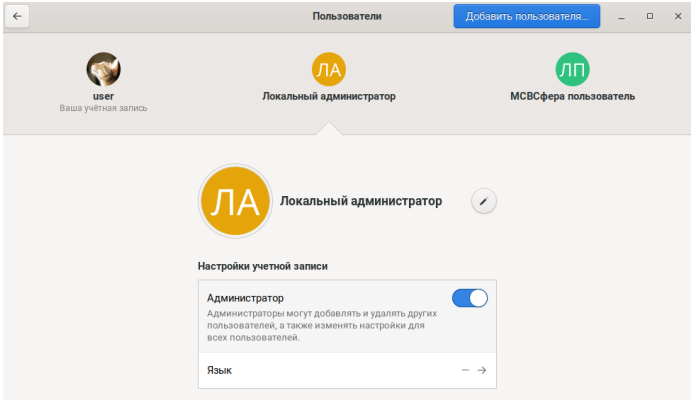
Обычный пользователь всегда может изменить свой пароль в приложении «**Настройки**» — «**Пользователи**» — «**Аутентификация и вход**», строка «**Пароль**». Для изменения пароля потребуется указать текущий пароль. После успешного указания всех требуемых данных, нажмите на кнопку «**Изменить**» в правом верхнем углу.



Администратор может менять как свой пароль, так и пароли обычных пользователей.

Изменение типа учётной записи

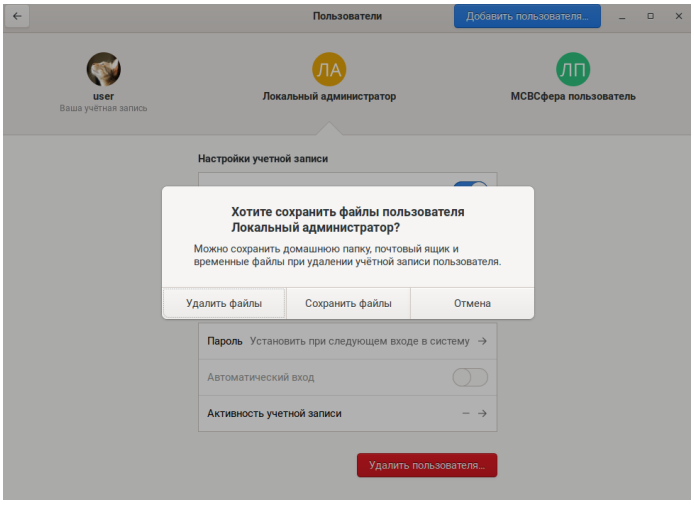
Изменение типа учётной записи доступно только для администратора. Для изменения типа учётной записи перейдите в раздел «**Пользователи**» приложения «**Настройки**» любым удобным способом. Затем передвиньте переключатель в строке «**Администратор**» требуемое положение.



Удаление пользователя

Удаление учётной записи доступно только для администратора, также нельзя удалить учётную запись, если она является единственной на этом устройстве.

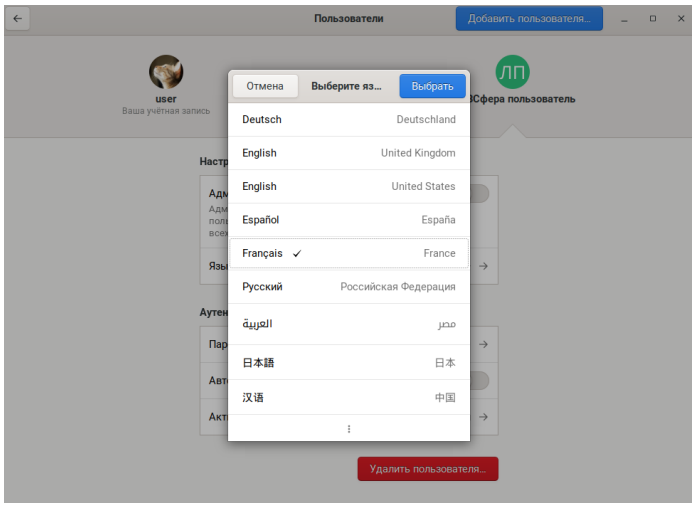
Для удаления учётной записи перейдите в раздел «**Пользователи**» приложения «**Настройки**» любым удобным способом. Затем нажмите на кнопку «**Удалить пользователя**», выберите сохранять ли файлы удаляемого пользователя и подтвердите своё решение.



Изменение языка для пользователя

Изменение языка доступно только для администратора.

Для изменения языка перейдите в раздел «**Пользователи**» приложения «**Настройки**» любым удобным способом. Выберите «**Язык**», откроется окно выбора языка. Выберите необходимый язык и нажмите «**Выбрать**» в правом верхнем углу.

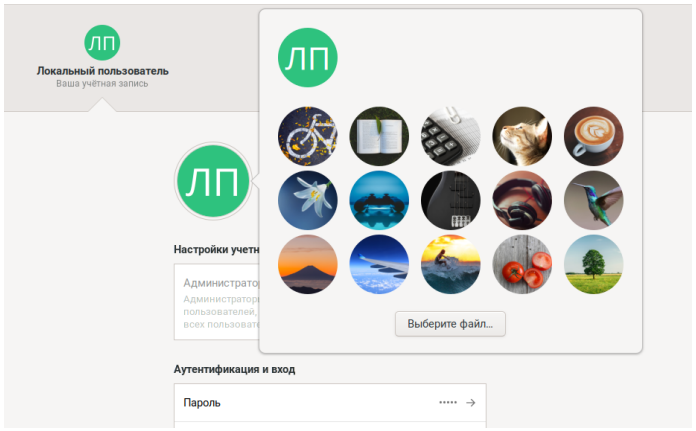


Изменение имени и графического представления для пользователя

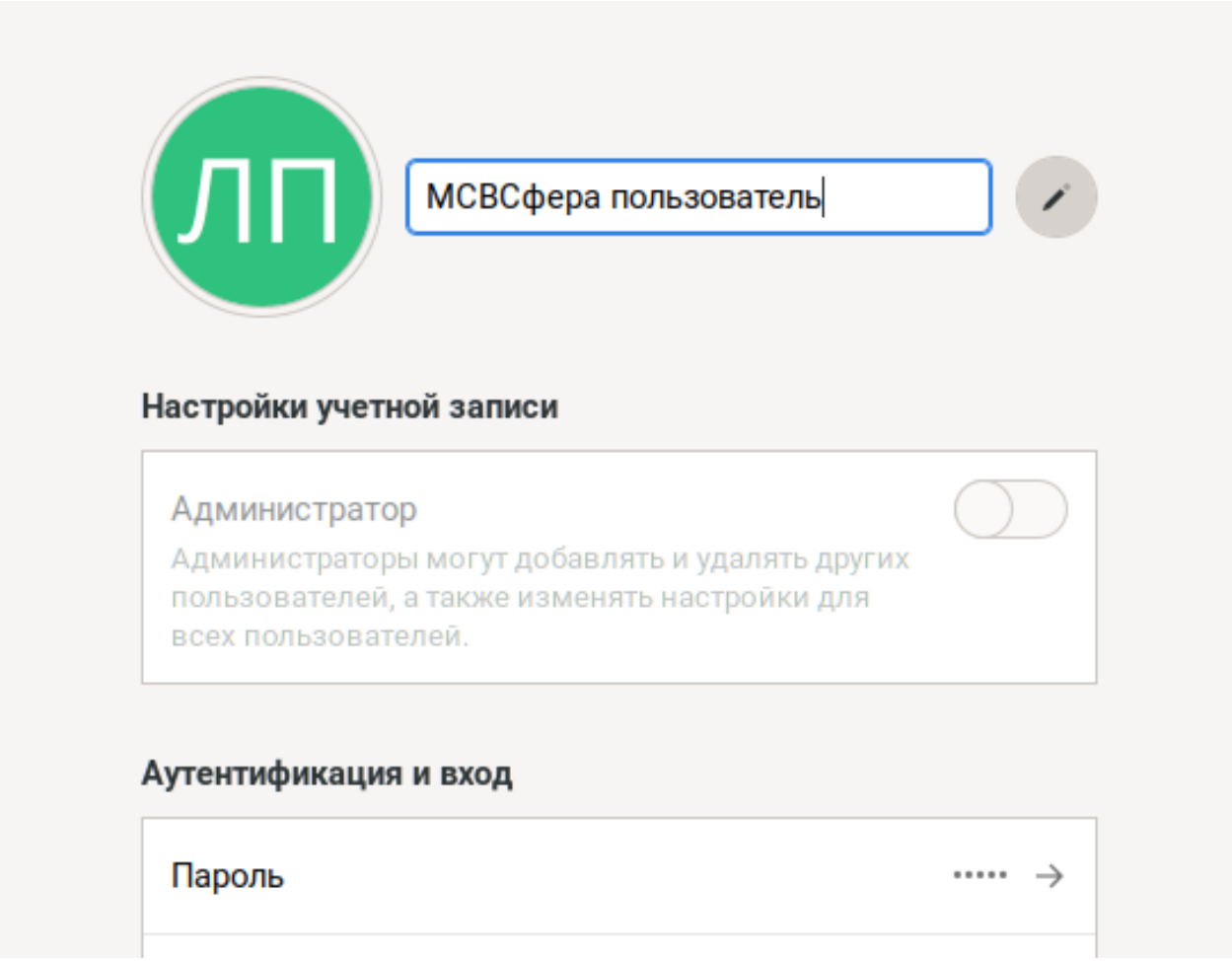
Пользователь также может изменить имя и графическое представление (изображение). Обычный пользователь может изменить только свои настройки, администратор может изменять настройки других пользователей.

Для изменения имени и графического представления перейдите в раздел «**Пользователи**» приложения «**Настройки**» любым удобным способом.

Для изменения графического представления нажмите на текущее изображение, затем выберите изображение из предлагаемых или из файла на компьютере.



Для изменения имени нажмите на изображение «**карандаша**», текущее имя пользователя станет доступным для изменения. При изменении имени пароль, а также имя домашней папки не изменятся.

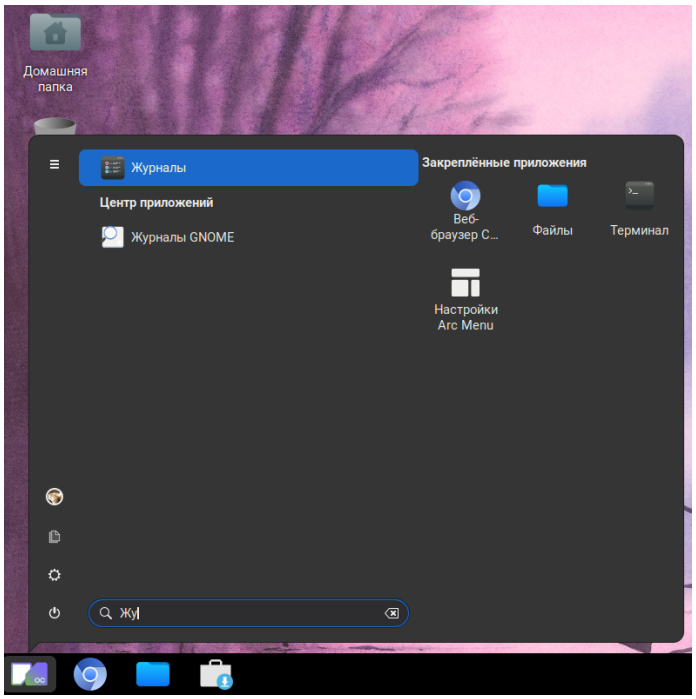


Для выхода без сохранения изменений просто закройте окно приложения «**Настройки**».

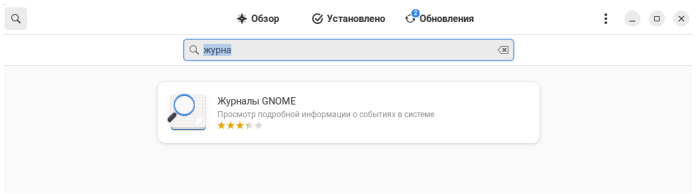
Просмотр системных журналов

Введение

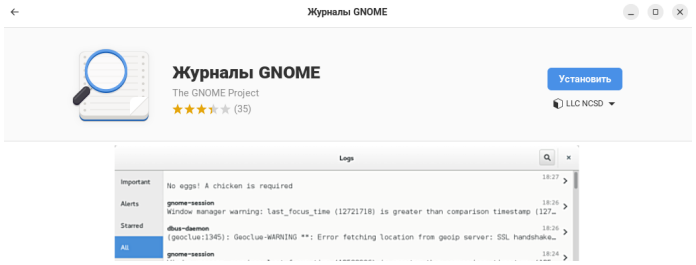
Просмотр системных журналов в графическом интерфейсе ОС МСВСфера производится в приложении «**Журналы GNOME**». Перейти в приложение «**Журналы GNOME**» вы можете из главного меню, набрав в строке поиска «журналы» и нажав на приложение правой кнопкой мыши.



Если приложение не установлено по умолчанию, его можно установить из «**Центра приложений**». Для этого перейдите в «**Центр приложений**» любым удобным способом и в строке поиска наберите «Журналы GNOME», нажмите правой клавишей мыши на найденное приложение, вы перейдёте на страницу приложения.



Далее нажмите на кнопку «**Установить**». После окончания установки приложение появится в главном меню.

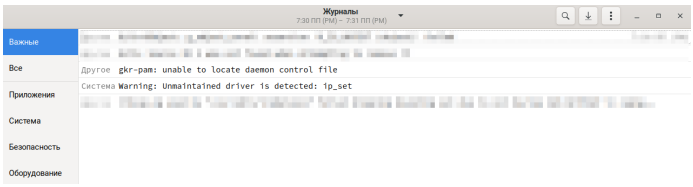


Записи в системном журнале

После запуска приложения вы попадёте на главную страницу. Слева показаны категории, справа список событий.

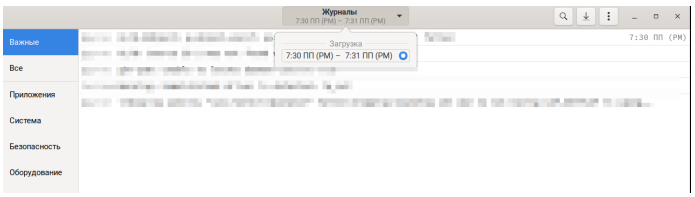
Доступны следующие категории:

- «Важные»;
- «Все»;
- «Приложения»;
- «Система»;
- «Безопасность»;
- «Оборудование».



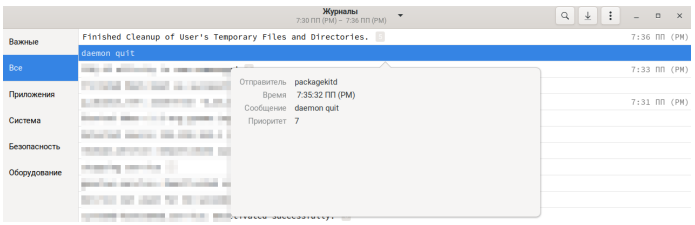
По умолчанию на главной странице отображаются события из категории «Важные».

Вы также можете выбрать загрузку, для которой показывать системный журнал, для этого нажмите на стрелку рядом с надписью «**Журналы**» в верхей части экрана.



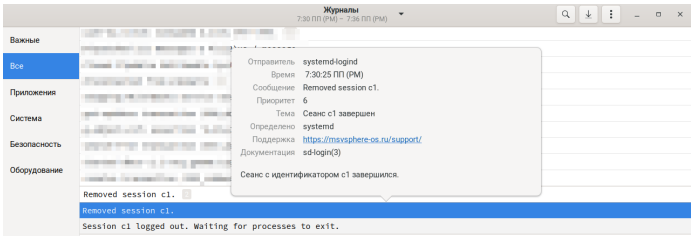
Просмотр информации о событии

Для просмотра информации о событии, нажмите на него левой кнопкой мыши.



Если в одно событие попадает несколько сообщений, то рядом с событием отображается цифра, соответствующая числу сообщений. При нажатии на такое

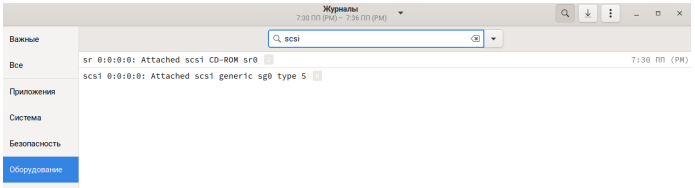
событие сначала раскрывается список сообщений, а затем, по нажатию правой клавишей мыши на сообщение, отображается подробная информация.



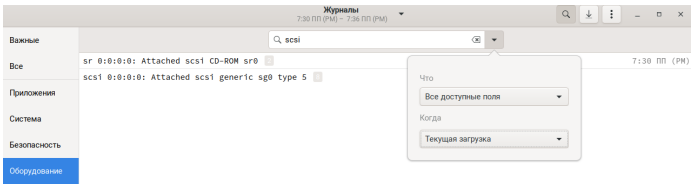
Поиск определённого события

Для поиска определённого события вы можете вызвать строку поиска, нажав на соответствующее изображение в правом верхнем углу экрана.

В строке поиска укажите ключевое слово, по которому будет осуществляться поиск — это может быть тип события, например, «error» или «warning», или же идентификатор устройства, например, «scsi».

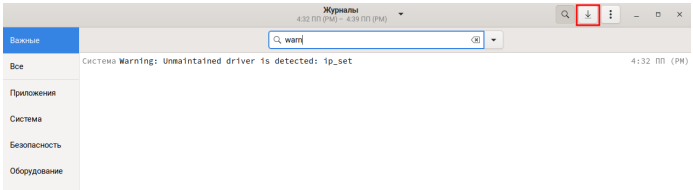


Для выбора дополнительных параметров поиска нажмите на стрелочку рядом с поисковой строкой, откроется выпадающее меню, в котором вы можете выбрать дополнительные параметры фильтрации — поле журнала и диапазон меток времени.



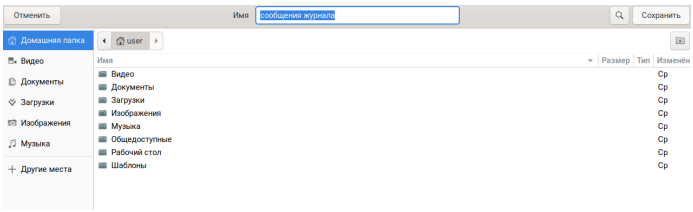
Экспорт журнала

Вы можете сохранить журнал для дальнейшего анализа или для других целей, предусмотренных политикой безопасности вашей компании, для этого нажмите на соответствующую иконку в правом верхнем углу (выделена красным на снимке экрана).



Обратите внимание, что если перед экспортом вы выполняли поиск, то экспорт будет выполнен только для найденных событий.

Укажите имя файла и куда его сохранять и нажмите на кнопку «**Сохранить**» в правом верхнем углу экрана.



Файл будет сохранён в указанное место, открыть его вы можете обычным текстовым редактором.

Создание защищённых каналов связи (VPN)

Создание защищённых VPN-туннелей, использующих контроль заголовков IP-пакетов в соответствии с ГОСТ Р 34.12-2015

Введение

Данный раздел описывает процедуру настройки VPN-туннеля на базе OpenVPN с контролем заголовков IP-пакетов в соответствии с ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры».

Все команды, описанные в данном разделе, необходимо выполнять с привилегиями системного администратора (`root`).

Предварительные требования

На сервере и на клиентских компьютерах необходимо установить последнюю версию криптографических политик и пакет `openssl-gost-engine`, добавляющий поддержку ГОСТ алгоритмов в OpenSSL:

```
$ dnf install openssl-gost-engine
$ dnf upgrade crypto-policies
```

Затем подключите модуль `TLSGOST` к текущей политике:

```
# отобразить на экран название текущей политики
$ update-crypto-policies --show
DEFAULT

# добавить к политике "DEFAULT" модуль "TLSGOST"
$ update-crypto-policies --set DEFAULT:TLSGOST
```

Дополнительную информацию о криптографических политиках и модулях вы можете получить из раздела «gost-alg».

Настройка сервера OpenVPN

Установка сервера OpenVPN

На компьютере, который будет выполнять роль сервера OpenVPN, установите соответствующий пакет:

```
$ dnf install openvpn
```

Настройка удостоверяющего центра

Для настройки сервера OpenVPN с авторизацией по сертификатам TLS необходимо настроить собственный удостоверяющий центр (англ. Certification authority, CA), основной функцией которого является управление сертификатами сервера и клиентов OpenVPN, в том числе их создание и отзыв.

В данном примере в качестве рабочего каталога для удостоверяющего центра будет использоваться каталог `/root/CA`. Создайте необходимую структуру каталогов, а также файлы `index.txt`, `serial` и `crlnumber`, требуемые для работы центра:

```
$ mkdir -p /root/CA/{certs,crl,newcerts,private}
$ touch /root/CA/index.txt
$ echo 1000 > /root/CA/serial
$ echo 01 > /root/CA/crlnumber
```

Создайте конфигурационный файл для OpenSSL:

```
$ cat > /root/CA/openssl.conf << EOF
[ca]
default_ca          = CA_default

[CA_default]
dir                 = /root/CA
certs               = \$dir/certs
crl_dir             = \$dir/crl
new_certs_dir       = \$dir/newcerts
database            = \$dir/index.txt
serial              = \$dir/serial
crlnumber           = \$dir/crlnumber
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
private_key         = \$dir/private/ca.key
certificate         = \$dir/certs/ca.crt
crl                 = \$dir/crl/ca.crl
policy              = policy_strict
default_days        = 365
default_crl_days    = 30
default_md          = gost12_512
preserve            = no

[policy_strict]
countryName         = match
stateOrProvinceName = match
organizationName    = match
commonName          = supplied

[req]
default_bits        = 2048
prompt              = no
encrypt_key         = no

[v3_ca]
basicConstraints    = CA:TRUE
keyUsage            = critical,keyCertSign,cRLSign

[server]
basicConstraints    = CA:FALSE
keyUsage            = digitalSignature,keyAgreement
extendedKeyUsage    = serverAuth

[client]
basicConstraints    = CA:FALSE
keyUsage            = digitalSignature,keyAgreement
extendedKeyUsage    = clientAuth
EOF
```

Создайте секретный ключ удостоверяющего центра, защищённый паролем:

```
$ openssl genpkey -aes256 -algorithm gost2012_512 -pkeyopt paramset:A \
    -out /root/CA/private/ca.key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

В тестовом окружении вы можете использовать секретный ключ без пароля — для этого из приведённой выше команды необходимо убрать аргумент `-aes256`. Однако, в условиях промышленной эксплуатации ключ обязательно нужно защищать паролем и хранить в надёжном месте.

Далее создайте самоподписанный сертификат удостоверяющего центра:

```
$ openssl req -x509 -new -config /root/CA/openssl.conf \
    -extensions v3_ca -key /root/CA/private/ca.key \
    -subj '/C=RU/ST=Moscow/O=MyVPN/CN=MyCA' \
    -days 3650 -out /root/CA/certs/ca.crt
Enter pass phrase for /root/CA/private/ca.key:
```

Перед запуском установите для полей аргумента `-subj` значения, соответствующие вашей организации: `C=RU` — страна, `ST=Moscow` — область, `O=MyVPN` — название организации, `CN=MyCA` — название удостоверяющего центра. Параметр `-days` определяет количество дней, в течение которых сертификат будет считаться действительным. В этом примере сертификат выпускается на 3650 дней (10 лет).

Затем, создайте список отозванных сертификатов:

```
$ openssl ca -config /root/CA/openssl.conf -gencrl \
    -out /root/CA/crl/ca.crl -md gost12_512 crldays 30
Using configuration from /root/CA/openssl.conf
Enter pass phrase for /root/CA/private/ca.key:
```

На этом процедуру создания удостоверяющего центра можно считать завершённой.

Конфигурация и запуск сервера OpenVPN

Создайте файл секретного ключа для сервера OpenVPN:

```
$ openssl genpkey -algorithm gost2012_512 -pkeyopt paramset:A \
    -out /root/CA/server.key
```

Создайте CSR запрос (Certificate Signing Request) на выпуск сертификата для сервера OpenVPN:

```
$ openssl req -new -config /root/CA/openssl.conf \
    -key /root/CA/server.key -subj '/C=RU/ST=Moscow/O=MyVPN/CN=server' \
    -out /root/CA/server.csr -md_gost12_512
```

Перед запуском установите для полей аргумента `-subj` значения, соответствующие вашей организации: `C=RU` — страна, `ST=Moscow` — область, `O=MyVPN` — название организации, `CN=server` — название сервера OpenVPN. В результате выполнения команды будет создан файл CSR запроса */root/CA/server.csr*, который затем будет использован для выпуска сертификата сервера OpenVPN удостоверяющим центром.

Для выпуска сертификата сервера OpenVPN выполните следующую команду:

```
$ openssl ca -config /root/CA/openssl.conf -in /root/CA/server.csr \
    -out /root/CA/server.crt -extensions server -md gost12_512 -batch
Using configuration from /root/CA/openssl.conf
Enter pass phrase for /root/CA/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'RU'
stateOrProvinceName   :ASN.1 12:'Moscow'
organizationName      :ASN.1 12:'MyVPN'
commonName            :ASN.1 12:'server'
Certificate is to be certified until May 15 15:19:27 2026 GMT (365 days)

Write out database with 1 new entries
Database updated
```

Выпущенный ключ будет записан в файл `/root/CA/server.crt`.

Затем необходимо сгенерировать файл с параметрами безопасности Диффи-Хеллмана для создания безопасного TLS соединения:

```
$ openssl dhparam -out /root/CA/dh2048.pem 2048
Generating DH parameters, 2048 bit long safe prime
...
```

Скопируйте созданные файлы в рабочий каталог сервера OpenVPN и установите безопасные права на файл секретного ключа:

```
$ cp -f /root/CA/certs/ca.crt \
    /root/CA/server.key \
    /root/CA/server.crt \
    /root/CA/dh2048.pem \
    /root/CA/crl/ca.crl \
    /etc/openvpn/server/

$ chmod 600 /etc/openvpn/server/server.key
```

где:

- `/root/CA/certs/ca.crt` — публичный сертификат удостоверяющего центра;
- `/root/CA/server.key` — секретный ключ сервера OpenVPN;
- `/root/CA/server.crt` — публичный сертификат сервера OpenVPN;
- `/root/CA/dh2048.pem` — параметры безопасности Диффи-Хеллмана;
- `/root/CA/crl/ca.crl` — список отозванных клиентских сертификатов.

Далее, создайте конфигурационный файл сервера OpenVPN:

```
$ cat > /etc/openvpn/server/server.conf << EOF
mode server
topology subnet
;local 192.168.10.38
port 1194
proto udp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
dh /etc/openvpn/server/dh2048.pem
crl-verify /etc/openvpn/server/ca.crl

server 10.10.4.0 255.255.255.0
;push "route 192.168.10.0 255.255.255.0"
;push "dhcp-option DNS 192.168.10.1"

keepalive 10 120
cipher kuznyechik-cbc
user openvpn
group openvpn
persist-key
persist-tun
verb 3
explicit-exit-notify 1
data-ciphers kuznyechik-cbc
auth id-tc26-gost3411-12-512
EOF
```

По умолчанию сервер OpenVPN принимает запросы со всех сетевых интерфейсов, раскомментируйте директиву `local` и укажите ей в качестве значения IP-адрес интерефейса, на котором должен принимать подключения сервер.

Далее, разрешите доступ к порту `1194`, на котором принимает подключения OpenVPN, в настройках брандмауэра:

```
$ firewall-cmd --permanent --add-service=openvpn
$ firewall-cmd --reload
```

После этого активируйте и запустите службу сервера OpenVPN:

```
$ systemctl enable --now openvpn-server@server.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /usr/
↪lib/systemd/system/openvpn-server@.service.
```

Проверить статус службы можно с помощью следующей команды:

```
$ systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/usr/lib/systemd/system/openvpn-server@.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-05-15 17:07:57 UTC; 5s ago
...
```

В диагностических целях системный журнал службы сервера OpenVPN можно просмотреть с помощью следующей команды:

```
$ sudo journalctl -u openvpn-server@server.service
```

Создание клиентских сертификатов

Для подключения к серверу OpenVPN для каждого клиента необходимо создать собственный секретный ключ и сертификат, все операции проводятся на сервере, выполняющем роль удостоверяющего центра.

Создайте файл сектретного ключа `/root/CA/client1.key`:

```
$ openssl genpkey -algorithm gost2012_512 -pkeyopt paramset:A \
    -out /root/CA/client1.key
```

Создайте CSR запрос на выпуск сертификата:

```
$ openssl req -new -config /root/CA/openssl.conf -key /root/CA/client1.key \
    -subj "/C=RU/ST=Moscow/O=MyVPN/CN=client1" \
    -out /root/CA/client1.csr -md_gost12_512
```

Перед запуском команды внесите необходимые изменения в поля аргумента `-subj`.

Создайте клиентский сертификат:

```
$ openssl ca -config /root/CA/openssl.conf -in /root/CA/client1.csr \
    -out /root/CA/client1.crt -extensions client -md gost12_512 -batch
Using configuration from /root/CA/openssl.conf
Enter pass phrase for /root/CA/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'RU'
stateOrProvinceName   :ASN.1 12:'Moscow'
organizationName      :ASN.1 12:'MyVPN'
commonName            :ASN.1 12:'client1'
Certificate is to be certified until May 15 18:54:10 2026 GMT (365 days)

Write out database with 1 new entries
Database updated
```

В результате выполнения команды будет создан файл `/root/CA/client1.crt`.

Затем, необходимо сгенерировать конфигурационный файл (в этом примере — `client1.ovpn`) для последующей передачи на клиентский компьютер:

```
$ cat > client1.ovpn << EOF
client
dev tun
proto udp
remote 192.168.10.38 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
cipher kuznyechik-cbc
verb 3
data-ciphers kuznyechik-cbc
auth id-tc26-gost3411-12-512

<ca>
$(cat /root/CA/certs/ca.crt)
</ca>

<cert>
$(openssl x509 -in /root/CA/client1.crt)
</cert>

<key>
$(cat /root/CA/client1.key)
</key>
EOF
```

Замените `192.168.10.38` на реальный IP-адрес вашего сервера OpenVPN. Созданный конфигурационный файл необходимо передать на клиентский компьютер. Используйте только защищённые каналы связи поскольку в этом файле находится в том числе и секретный ключ доступа к VPN.

Настройка клиента OpenVPN

Установка клиента OpenVPN

На компьютере, который будет выполнять роль клиента OpenVPN, установите соответствующий пакет:

```
$ sudo dnf install openvpn
```

Если этот компьютер является графической рабочей станцией, установите также расширение для NetworkManager, которое позволяет настраивать подключение к OpenVPN через графический интерфейс:

```
$ sudo dnf install NetworkManager-openvpn-gnome
```

Настройка клиента OpenVPN в режиме командной строки

Для настройки клиента OpenVPN в режиме командной строки скопируйте созданный ранее файл `client1.ovpn` в каталог `/etc/openvpn/client/` под именем `client.conf` и установите для него безопасные права доступа:

```
$ sudo cp client1.ovpn /etc/openvpn/client/client.conf
$ sudo chown root:root /etc/openvpn/client/client.conf
$ sudo chmod 600 /etc/openvpn/client/client.conf
```

Следующая команда активирует и запустит службу клиента OpenVPN:

```
$ sudo systemctl enable --now openvpn-client@client.service
```

Посмотреть статус службы можно следующим образом:

```
$ sudo systemctl status openvpn-client@client.service
● openvpn-client@client.service - OpenVPN tunnel for client
   Loaded: loaded (/usr/lib/systemd/system/openvpn-client@.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-05-16 00:50:08 MSK; 16min ago
```

После запуска службы и успешного подключения к VPN-серверу в системе появится новый сетевой интерфейс `tun0`:

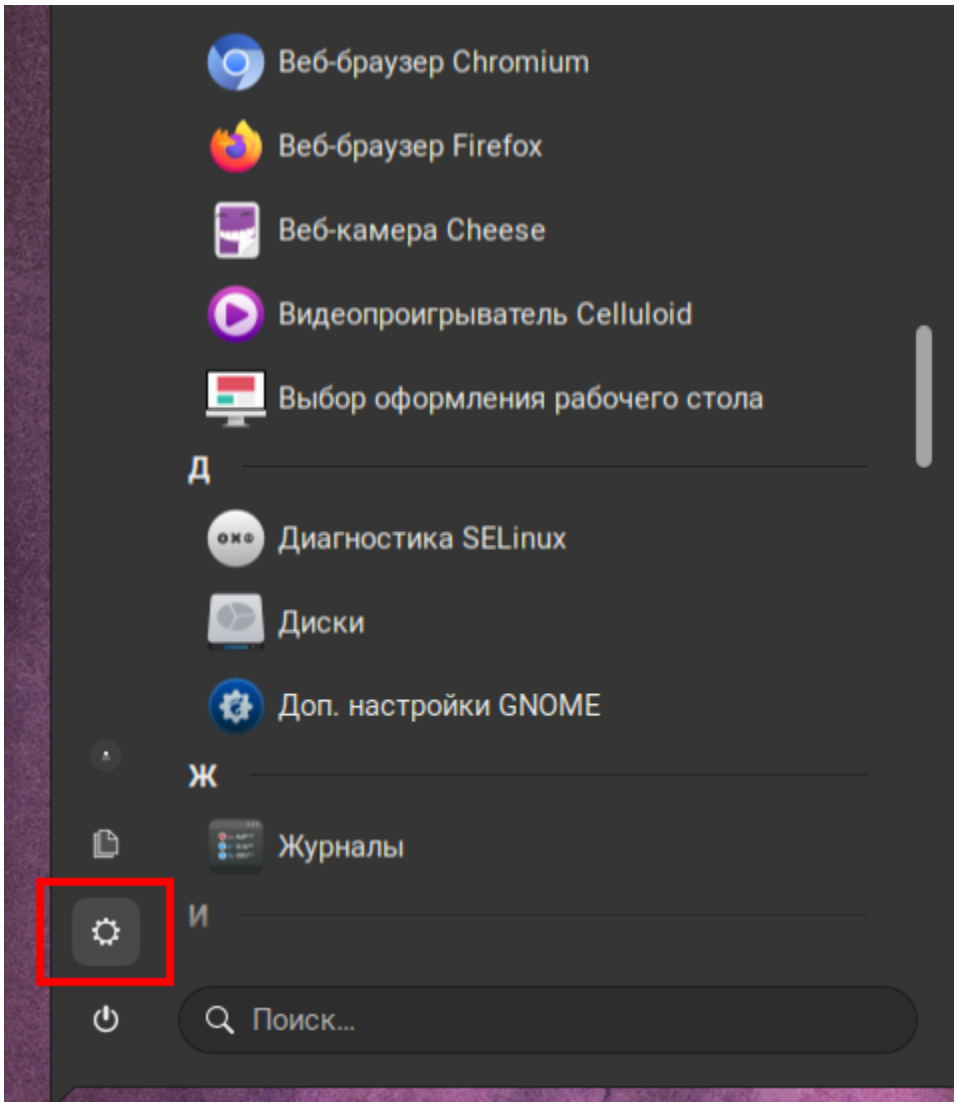
```
...
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group␣
↪default qlen 500
  link/none
  inet 10.10.4.2/24 scope global tun0
     valid_lft forever preferred_lft forever
  inet6 fe80::640:643d:d512:52ce/64 scope link stable-privacy
     valid_lft forever preferred_lft forever
```

В диагностических целях системный журнал клиента OpenVPN можно просмотреть с помощью следующей команды:

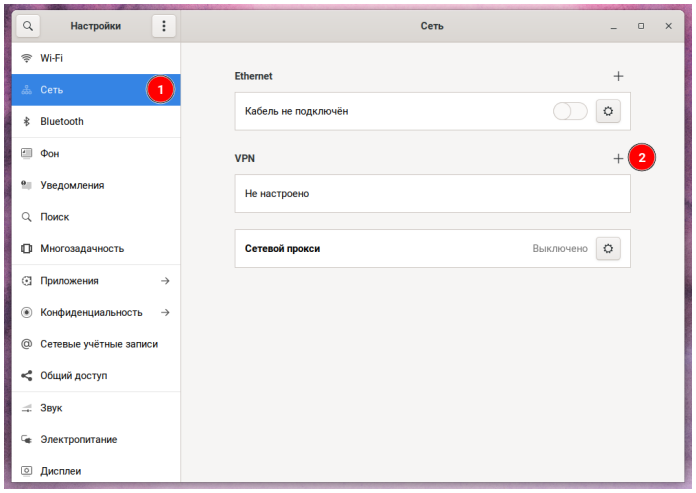
```
$ sudo journalctl -u openvpn-client@client.service
```

Настройка клиента OpenVPN в графическом режиме

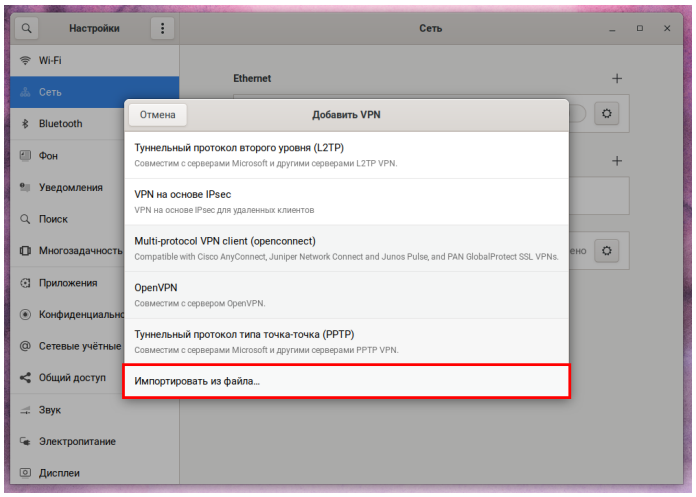
Для настройки подключения к OpenVPN через графический интерфейс откройте главное меню системы и запустите приложение «Настройки» (помечено красной рамкой на снимке экрана ниже):



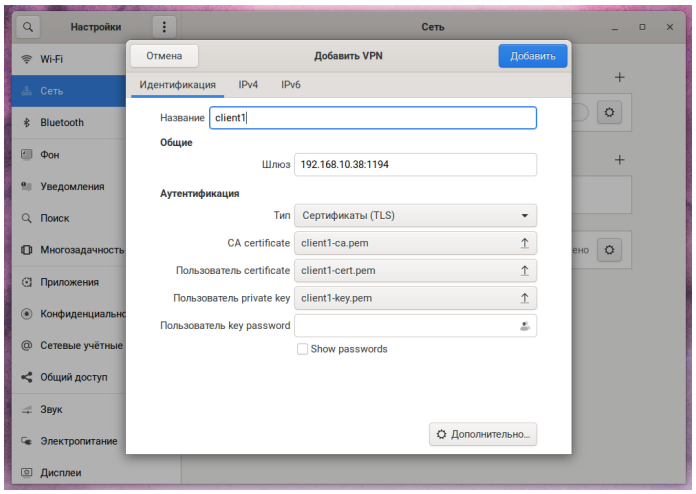
В левом меню выберите пункт «Сеть» (отмечен цифрой 1 на снимке экрана ниже), после этого откроется панель настройки сетевых подключений, где в блоке «VPN» вам необходимо нажать на значок + (отмечен цифрой 2 на снимке экрана ниже):



В появившемся окне «Добавить VPN» нажмите на пункт «Импортировать из файла...», отмеченный красной рамкой на снимке экрана ниже:

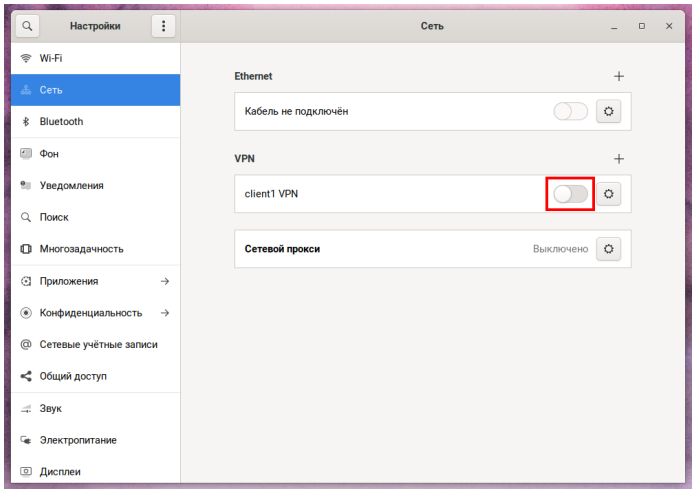


В открывшемся окне выбора файлов выберите ранее созданный файл `client1.ovpn` и нажмите кнопку «Открыть» — после этого появится окно добавления нового VPN-подключения, представленное на следующем снимке экрана:



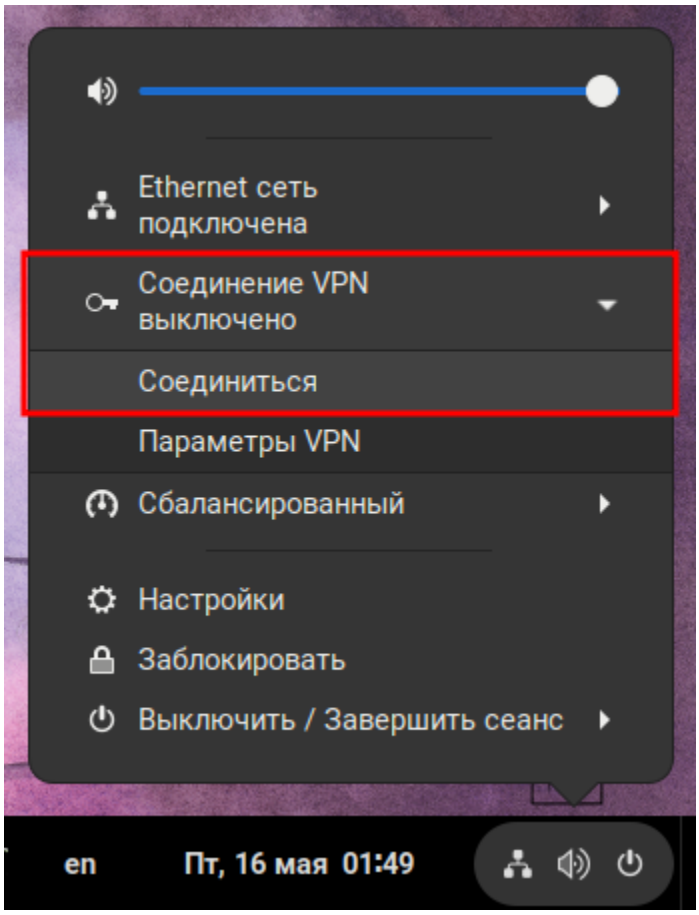
В случае необходимости вы можете изменить название подключения, IP-адрес или TCP-порт сервера. После завершения настройки нажмите на кнопку «Добавить».

После этого новое VPN-соединение появится в блоке «VPN» панели сетевых настроек:



Для подключения к VPN активируйте соответствующий переключатель, отмеченный красной рамкой на снимке экрана выше.

Также вы можете подключиться к ранее созданному VPN-соединению, нажав на группу иконок в правом нижнем углу экрана (справа от часов) и выбрав там пункт «Соединиться» в соответствующем блоке, отмеченном красной рамкой на снимке экрана:



Ограничение времени работы за компьютером

Введение

В состав операционной системы МСВСфера входит приложение **Timekpr-nExT**, которое предназначено для контроля и ограничения времени, проведённого пользователем за компьютером.

Установка

Для установки **Timekpr-nExT** выполните следующую команду:

```
$ sudo dnf install timekpr-next
```

Также установку можно выполнить с помощью «Центра приложений».

Компоненты системы

Служба timekpr

Основным компонентом **Timekpr-nExT** является системная служба `timekpr`, которая осуществляет учёт времени, проведённого пользователем за компьютером, и применение соответствующих ограничений.

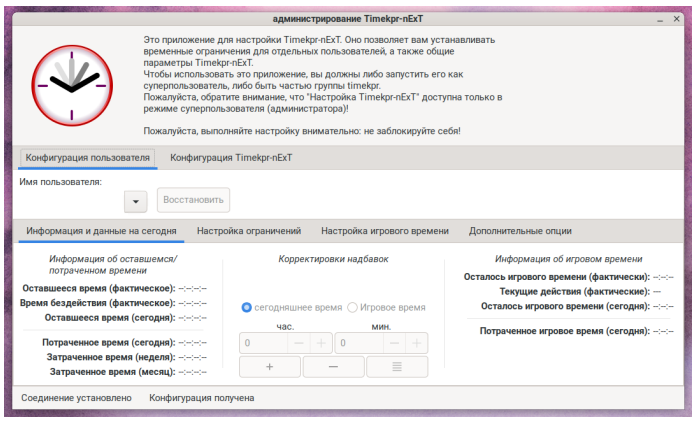
Сервис `timekpr` запускается автоматически после установки RPM-пакета и в процессе загрузки компьютера, проверить статус можно с помощью следующей команды:

```
$ systemctl status timekpr
● timekpr.service - Timekpr-nExT daemon service
     Loaded: loaded (/usr/lib/systemd/system/timekpr.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-05-27 10:11:37 MSK; 36s ago
       Docs: file:/etc/timekpr/timekpr.conf
   Main PID: 1606 (timekprd)
      Tasks: 3 (limit: 50155)
     Memory: 14.2M
        CPU: 98ms
     CGroup: /system.slice/timekpr.service
             └─1606 /usr/bin/python3 /usr/lib/python3.9/site-packages/timekpr/server/timekprd.py␣
↪/usr/bin/timekprd

мая 27 10:11:37 sphere-96-arm systemd[1]: Started Timekpr-nExT daemon service.
```

Панель управления

Для настройки системы используется графическая панель управления:



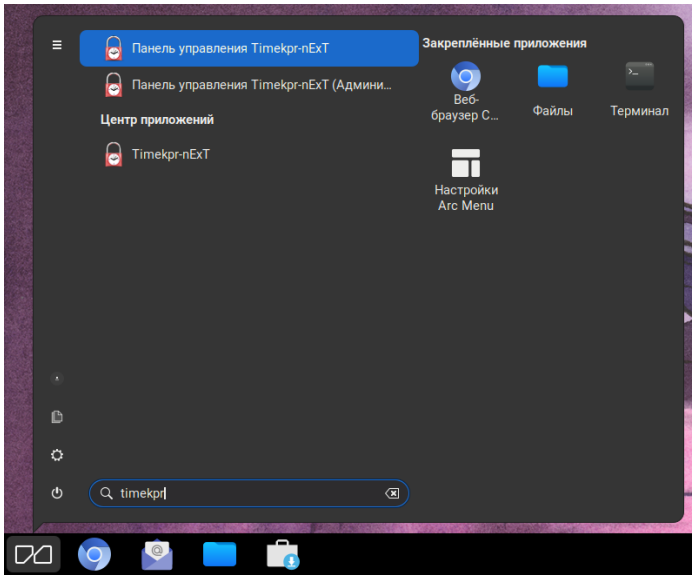
Для работы с панелью управления пользователь, выполняющий роль администратора системы учёта времени, должен либо обладать привилегиями системного администратора, либо быть участником системной группы `timekpr`. Второй вариант является предпочтительным с точки зрения информационной безопасности.

Для добавления пользователя в группу `timekpr` выполните следующую команду (замените `user` на реальное имя пользователя):

```
$ sudo gpasswd -a user timekpr
Добавление пользователя user в группу timekpr
```

После добавления в группу пользователю необходимо выйти из системы и зайти повторно для применения изменений.

Для запуска панели управления используйте пункт главного меню `Панель управления Timekpr-nExT`:



Либо выполните следующую команду в терминале:

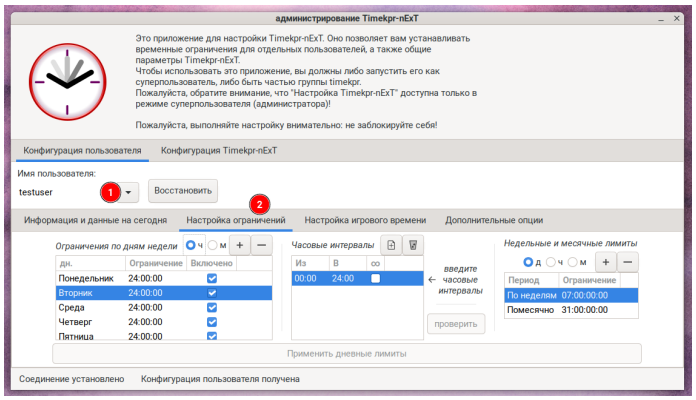
```
$ timekpra
```

Если по каким-то причинам вы не можете добавить пользователя в группу `timekpr`, то вам необходимо использовать пункт меню `Панель управления Timekpr-nExT (Администратор)` либо команду `sudo timekpra`. В таком случае система попросит вас ввести пароль.

Процедура настройки системы контроля времени с помощью панели управления описывается в следующих разделах.

Настройка ограничений времени

Для настройки ограничений рабочего времени запустите `Панель управления Timekpr-nExT`, выберите пользователя, для которого необходимо установить ограничения, в выпадающем списке пользователей (отмечен цифрой `1` на снимке экрана) и перейдите во вкладку `Настройка ограничений` (отмечена цифрой `2` на снимке экрана):



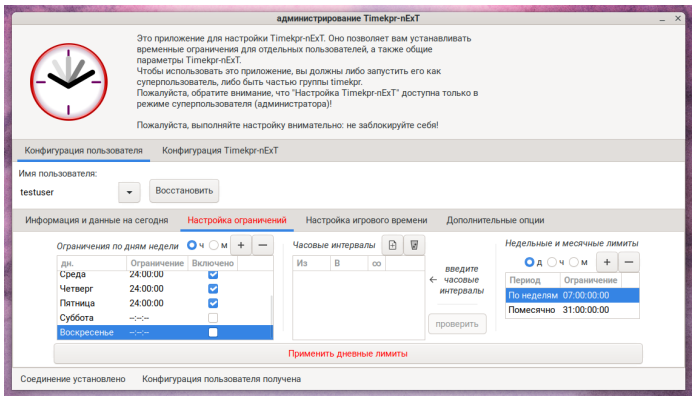
Для каждого пользователя можно установить следующие типы лимитов:

- *Ограничения по дням недели* — определяет в какие дни недели пользователь может использовать компьютер и суммарное количество времени, доступное для каждого дня;
- *Часовые интервалы* — определяет конкретные временные интервалы, в которых пользователь может использовать компьютер в течение дня;
- *Недельные и месячные лимиты* — определяет суммарное количество времени, доступное пользователю в течение недели и месяца.

Ограничения по дням недели

Лимит «Ограничения по дням недели» позволяет задать список дней, в которые пользователь может работать за компьютером, а также указать количество времени, доступное для каждого дня.

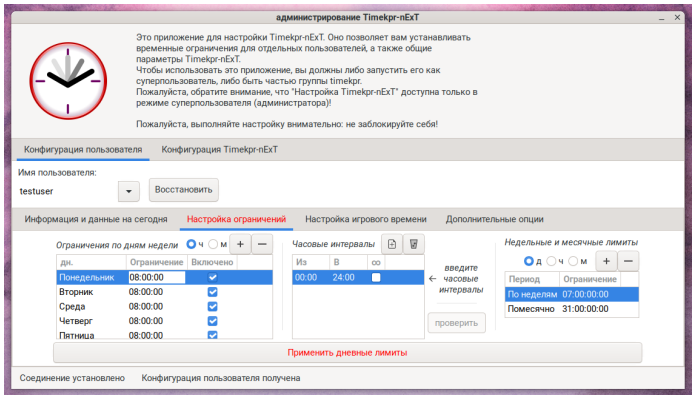
Для запрета работы в определённый день недели уберите флажок в колонке «Включено». На следующем снимке экрана в качестве примера установлен запрет на работу с компьютером в выходные дни:



Также вы можете ограничить суммарное количество времени для каждого дня: для этого выберите в таблице день недели, нажмите на соответствующую ячейку

в колонке «Ограничение» и укажите необходимое значение в формате `ЧЧ:ММ:СС` (`Часы:Минуты:Секунды`). Значение по умолчанию - `24:00:00`, что означает отсутствие ограничений по времени. В качестве альтернативного способа вы можете использовать переключатель `Ч/М` (`Часы/Минуты`) и кнопки `+/-` для изменения значения.

На следующем снимке экрана доступное время в будние дни ограничено восемью часами:

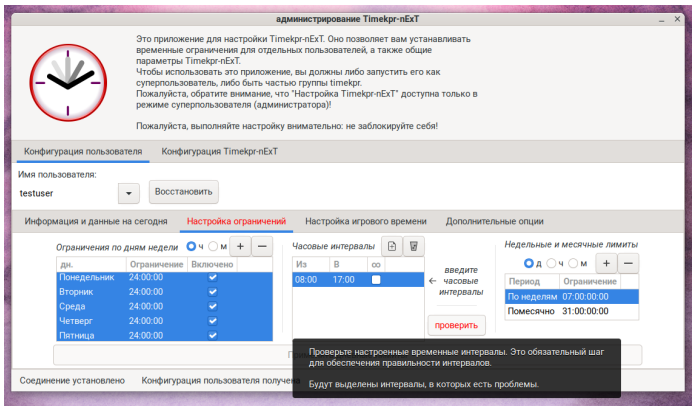


Часовые интервалы

С помощью часовых интервалов вы можете определить, в какие промежутки времени в течение дня пользователь может использовать компьютер. Настраивать их можно как для одного, так и для нескольких дней одновременно.

В конфигурации по умолчанию для всех дней задан один интервал с `00:00` до `24:00` — фактически это означает отсутствие ограничений.

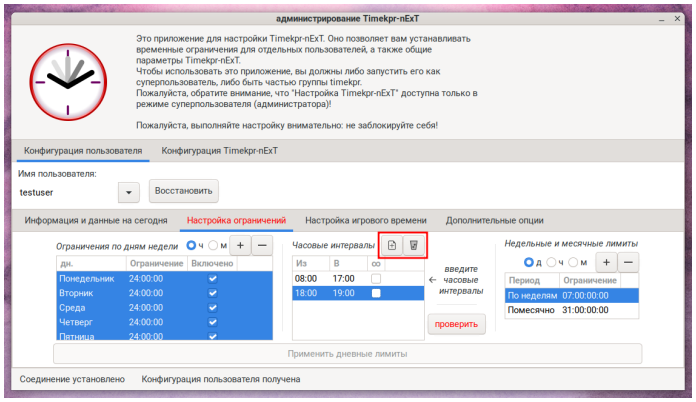
Для изменения существующего интервала выберите один или несколько дней в таблице «Ограничения по дням недели», затем в колонке «Из» установите начальное время, а в колонке «В» — конечное время интервала. На снимке экрана ниже показана установка временного интервала с `8:00` до `17:00` для всех будних дней:



После установки временного интервала вам необходимо нажать кнопку «проверить» для верификации введённых данных. В случае обнаружения ошибок проблемные

интервалы будут выделены, а в строке состояния программы отобразится описание ошибки.

Вы можете настроить несколько временных интервалов — используйте соответствующие кнопки, выделенные красной рамкой на снимке экрана ниже, для добавления нового и удаления выбранного интервала:



При настройке интервалов следует придерживаться следующих правил:

- интервалы не должны пересекаться;
- даже минутный перерыв между интервалами активирует ограничения;
- в течение одного часа не может начинаться и заканчиваться больше одного интервала.

Также следует упомянуть переключатель в колонке «∞» — его активация приведёт к тому, что выбранный интервал будет считаться свободным от ограничений и не будет учитываться в общем дневном лимите. Это может быть полезно, например, если ребёнок посещает онлайн курсы и время, проведённое на них, не должно приводить к уменьшению доступного для других активностей времени.

Недельные и месячные лимиты

С помощью блока недельных и месячных лимитов вы можете настроить максимально допустимое время использования компьютера за неделю или месяц. В конфигурации по умолчанию лимиты отсутствуют: для недельного ограничения установлено значение 7 дней, а для месячного — 31 день.

Для определения значений используется формат `Дни:Часы:Минуты:Секунды`. При вводе вы можете использовать сокращения, например: значение `6` будет преобразовано в `06:00:00:00`, а `6:12:30` — в `06:12:30:00`.

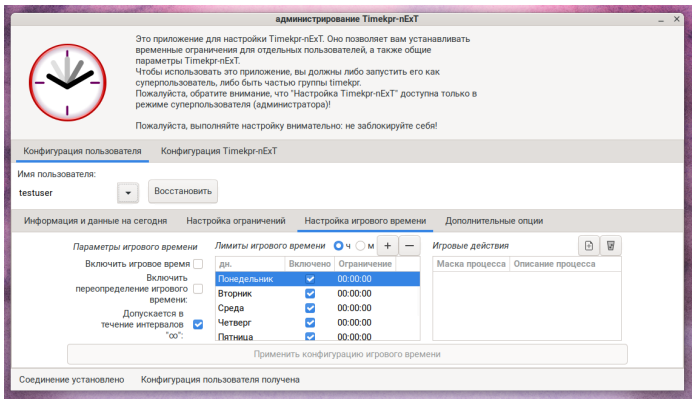
Недельные и месячные лимиты работают одновременно с дневными лимитами — система всегда будет использовать наименьшее из действующих ограничений.

Сохранение настроек

После установки необходимых ограничений нажмите кнопку «Применить дневные лимиты» для сохранения настроек. В случае успешного сохранения настроек программа отобразит подтверждающее сообщение в статусной строке — после этого вы можете перейти к настройкам лимитов для другого пользователя, либо закрыть программу.

Ограничение времени для отдельных процессов

В системе **Timekpr-nExT** также реализована функция ограничения времени использования отдельных процессов. Изначально этот модуль был разработан для ограничения времени, проведённого в играх детьми, однако может быть использован для контроля использования любых процессов. Настройки данной функции доступны во вкладке «Настройка игрового времени»:



В блоке «Параметры игрового времени» находится группа переключателей, отвечающая за общие настройки модуля:

- *Включить игровое время* — активирует функцию учёта времени использования отдельных процессов для выбранного пользователя. Также для работы этой функции потребуется включить соответствующий переключатель в *системных настройках* **Timekpr-nExT**;
- *Включить переопределение игрового времени* — активация этого переключателя отключает стандартные лимиты времени, учёт будет выполняться только для процессов, перечисленных в блоке «Игровые действия»;
- *Допускается в течение интервалов «∞»* — разрешает запуск процессов, определённых в блоке «Игровые действия» в интервалах, помеченных как свободные (∞) в блоке «Часовые интервалы» вкладки «Настройка ограничений».

В блоке «Лимиты игрового времени» настраиваются дни и часы, в которые пользователь сможет запускать процессы, определённые в блоке «Игровые действия».

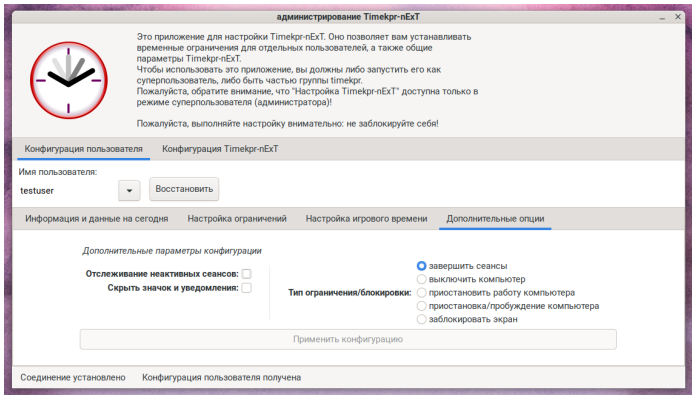
Процедура настройки идентична настройке *часовых интервалов* во вкладке «Настройка ограничений».

Блок «Игровые действия» является ключевым для данного модуля — в нём настраивается список процессов, для которых необходимо применять ограничение. В поле «Маска процесса» необходимо ввести название команды без пути и аргументов командной строки (используйте команды `top -c -d 1`, `htop` и т.п., чтобы получить информацию о запущенных в системе процессах). В поле «Описание процесса» вы можете добавить опциональный комментарий. Также вы можете включить *системную настройку* «Улучшенный мониторинг активности» — в этом случае будет активирована поддержка регулярных выражений и аргументов командной строки в поле «Маска процесса».

Для сохранения изменений нажмите кнопку «Применить конфигурацию игрового времени».

Дополнительные опции

Во вкладке «Дополнительные опции» находятся дополнительные параметры, которые могут быть настроены для каждого пользователя:



- *Отслеживание неактивных сеансов* — активирует учёт времени, даже когда экран компьютера заблокирован или пользователь работает в аппаратной консоли;
- *Скрыть значок и уведомления* — скрывает иконку клиентского приложения **Timekpr-nExT** на панели задач рабочего стола и отключает предупреждения о скором завершении сеанса;
- *Тип ограничения/блокировки* — определяет действие, которое должно быть выполнено по достижении лимита времени пользователем:
  - *завершить сеансы* — принудительно завершает сессию пользователя. В случае повторного входа в систему сессия будет также незамедлительно закрыта;

– *выключить компьютер* — выключает компьютер по достижении лимита. Используйте эту опцию с осторожностью, особенно в многопользовательских окружениях;

– *приостановить работу компьютера* — переводит компьютер в спящий режим. Пользовательская сессия при этом не завершается — при включении экран будет заблокирован. Если выход из спящего режима был осуществлён до сброса лимитов, компьютер будет повторно переведён в спящий режим;

– *приостановка/пробуждение компьютера* — действует так же, как и предыдущая опция, но компьютер будет автоматически выведен из спящего режима в указанном интервале этого же дня. Некоторое оборудование может быть несовместимо с этим режимом — как минимум требуется поддержка RTC (Real Time Clock) в BIOS/UEFI и именно программа **Timekpr-nExT** должна перевести компьютер в спящий режим. Если следующий период разблокировки лимитов не попадает в указанный в настройках опции интервал, компьютер не будет автоматически выводиться из спящего режима;

– *заблокировать экран* — по исчерпании лимита экран компьютера блокируется.

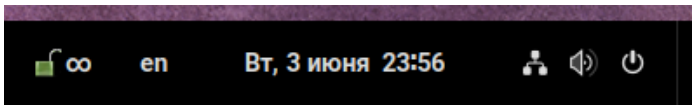
Последние три варианта не являются надёжными способами блокировки, а больше подходят для самоконтроля. В производственных условиях рекомендуется использовать варианты «завершить сеансы» или «выключить компьютер».

После внесения изменений нажмите кнопку «Применить конфигурацию» для сохранения настроек.

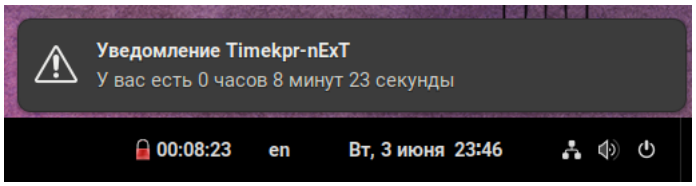
Пользовательское приложение

После установки и активации системы **Timekpr-nExT** у каждого пользователя на панели задач появится иконка приложения, отображающая текущий статус.

В случае отсутствия настроенных ограничений иконка будет иметь вид открытого замка с символом «∞»:

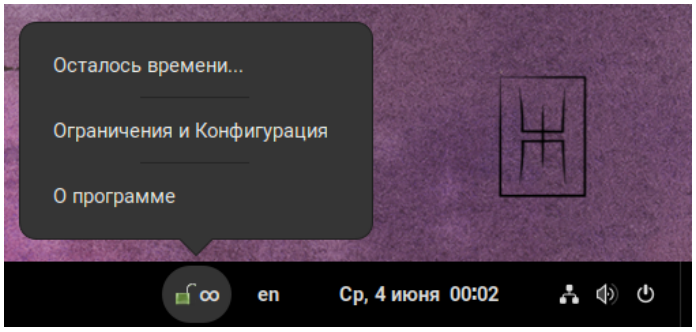


Если же лимиты активны, то иконка будет иметь вид закрытого замка и счётчика оставшегося времени до завершения сессии:

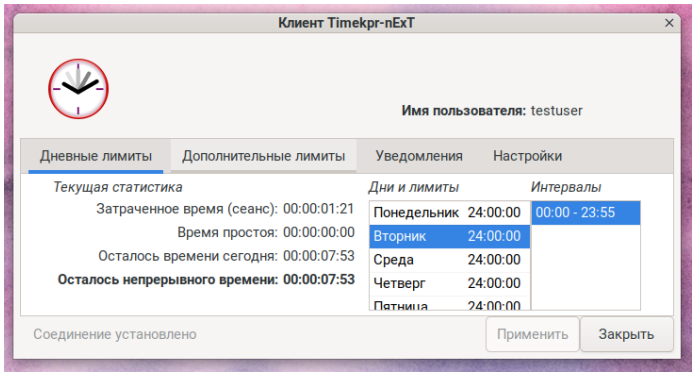


С заданной в настройках системы периодичностью пользователю будет отображаться всплывающее уведомление о количестве оставшегося времени до завершения сеанса.

Нажав на иконку приложения, пользователь вызовет контекстное меню:

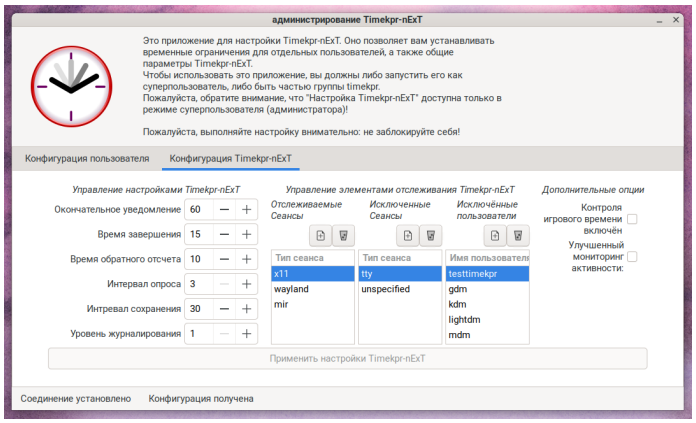


- *Осталось времени* — отобразит всплывающее уведомление о том, сколько времени осталось до блокировки сеанса;
- *О программе* — отобразит окно с информацией о версии программы **Timekpr-nExT**, её разработчиках и лицензии;
- *Ограничения и Конфигурация* — отобразит основное окно пользовательского приложения, в котором можно будет посмотреть статистику использования времени и настроить уведомления системы:



Системные настройки

Для доступа к системным настройкам **Timekpr-nExT** вам необходимо запустить *панель управления* с привилегиями системного администратора и перейти на вкладку «Конфигурация Timekpr-nExT»:



Для большинства сценариев использования конфигурация по умолчанию является оптимальной и не требует изменения.

Блок «Управление настройками Timekpr-nExT»:

- *Окончательное уведомление* — определяет, за сколько секунд до истечения выделенного времени пользователю будет показано финальное уведомление о предстоящем завершении сеанса;
- *Время завершения* — определяет, за сколько секунд до истечения выделенного времени будет применяться блокировка сеанса;
- *Время обратного отсчета* — определяет, за сколько секунд будет начат обратный отсчёт в реальном времени перед блокировкой сеанса;
- *Интервал опроса* — задаёт интервал в секундах, с которым система пересчитывает оставшееся время и анализирует сеансы пользователя;
- *Интервал сохранения* — задаёт интервал в секундах, с которым система будет сохранять статистику активности пользователя на диск;
- *Уровень журналирования* — задаёт уровень детализации сообщений в системном журнале службы:
  - *1* — стандартный (рекомендуется к использованию в производственной среде);
  - *2* — отладочный (включает дополнительную диагностическую информацию);
  - *3* — расширенный отладочный (включает вывод максимального количества диагностической информации, включая содержимое внутренних структур. Он используется, в основном, разработчиками системы).

Блок «Управление элементами отслеживания Timekpr-nExT»:

- *Отслеживаемые сеансы* — определяет типы сеансов, для которых будет выполняться учёт времени. В конфигурации по умолчанию отслеживаются

только сеансы «x11» (Xorg), «wayland» и «mir», что является достаточным для большинства сценариев использования. **Timekpr-nExT** предполагает, что существуют только эти типы сеансов и сеансы, указанные в списке «Исключённые сеансы»;

- *Исключенные сеансы* — список сеансов, для которых не будет осуществляться учёт времени. В конфигурации по умолчанию в этот список входят «tty» (аппаратные консоли, `Ctrl+Alt+F[1-7]`) и «unspecified» (все остальные типы сеансов, которые не входят в списки отслеживаемых и исключённых сеансов);
- *Исключенные пользователи* — список пользователей, для которых не должен осуществляться учёт времени. Как правило, это системные пользователи, от имени которых запускаются менеджеры входа в систему (GDM и т.п.).

Блок «Дополнительные опции»:

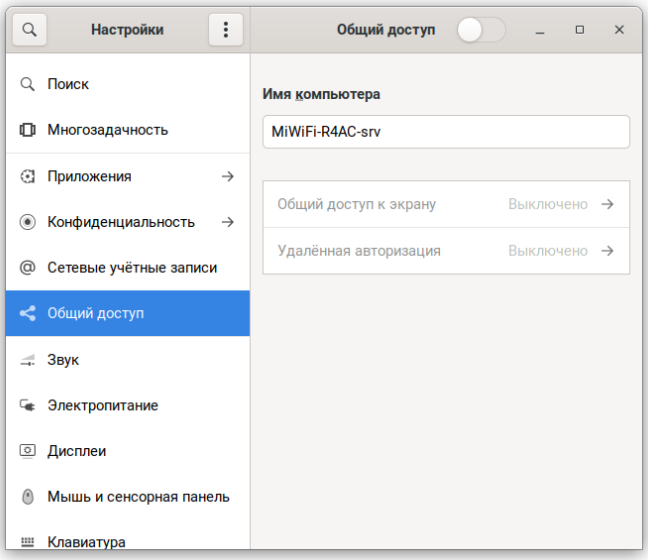
- *Контроль игрового времени включён* — включает или отключает функцию учёта игрового времени. Если этот переключатель выключен, то функция будет отключена для всех пользователей, даже если для отдельно взятого пользователя она активирована во вкладке «Настройка игрового времени»;
- *Улучшенный мониторинг активностей* — если эта опция включена, то функция учёта игрового времени будет отслеживать процессы по полной строке запуска процесса, включая аргументы командной строки, а если отключена, то отслеживание будет осуществляться только по названию команды. В некоторых случаях эта функция может быть полезна для отслеживания процессов, которые запускаются через интерпретаторы.

Для сохранения изменённых настроек нажмите кнопку «Применить настройки Timekpr-nExT».

Подключение к удалённому рабочему столу

Введение

В разделе настроек «Общий доступ» вы можете запустить встроенный в GNOME сервер VNC/RDP, который позволяет удалённо подключаться к рабочему пространству пользователя. В зависимости от версии операционной системы МСВСфера протокол подключения может отличаться.



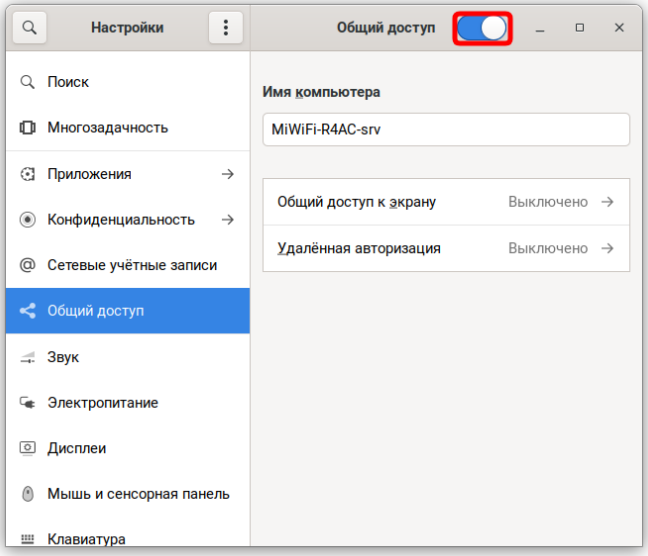
Установка

Установите пакет `gnome-remote-desktop`, для этого выполните следующую команду в «Терминале»:

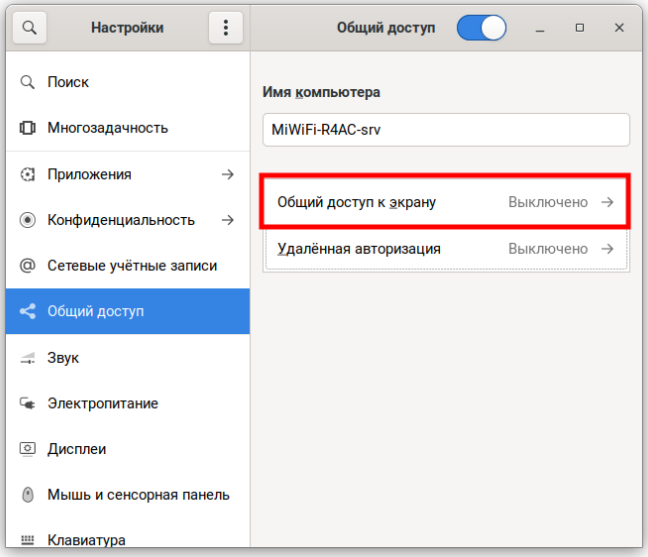
```
$ sudo dnf install gnome-remote-desktop
```

Использование раздела «Общий доступ»

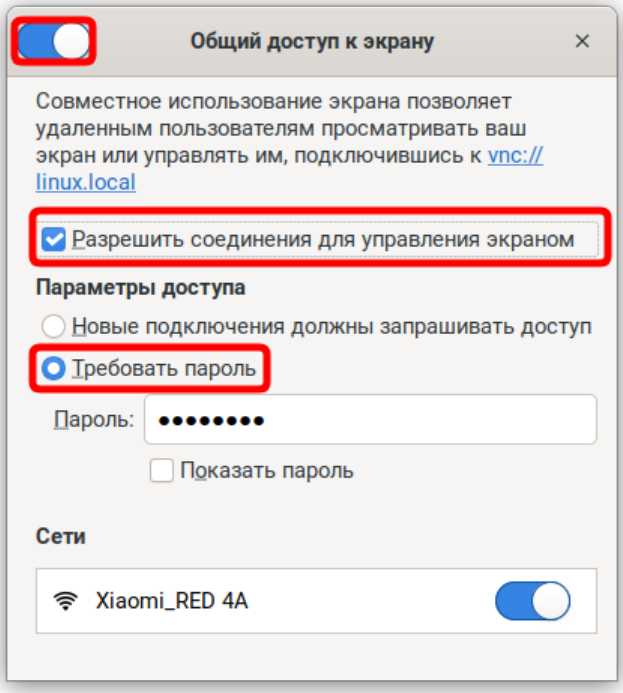
В ОС МСВСфера 9 перейдите в настройки и найдите раздел «Общий доступ», затем включите его.



После включения перейдите во вкладку «Общий доступ к экрану».



Включите функцию «Общий доступ к экрану» с помощью переключателя. Если вы хотите, чтобы пользователь имел доступ к управлению вашим экраном, поставьте галочку в строке «Разрешить соединения для управления экраном». Также при подключении рекомендуется использовать пароль, для этого выберите «Требовать пароль» и укажите его в специальном поле.



Чтобы удалённое подключение к рабочему столу стало возможным, откройте порт `5900` и перезагрузите `firewall`:

```
$ sudo firewall-cmd --permanent --add-port=5900/tcp

$ sudo firewall-cmd --reload
```

Предупреждение

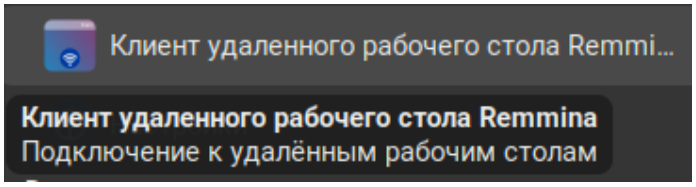
Существует проблема подключения к удалённой рабочей машине с помощью `vncviewer`. Она заключается в невозможности согласовать тип механизма безопасности, используемого в `gnome-remote-desktop`. Если вы хотите использовать только `vncviewer`, то в качестве временного решения вы можете отключить шифрование в `gnome-remote-desktop` с помощью следующей команды: `$ gsettings set org.gnome.desktop.remote-desktop.vnc encryption "['none']"`.

Удалённое подключение к ОС МСВСфера 9 с помощью Remmina

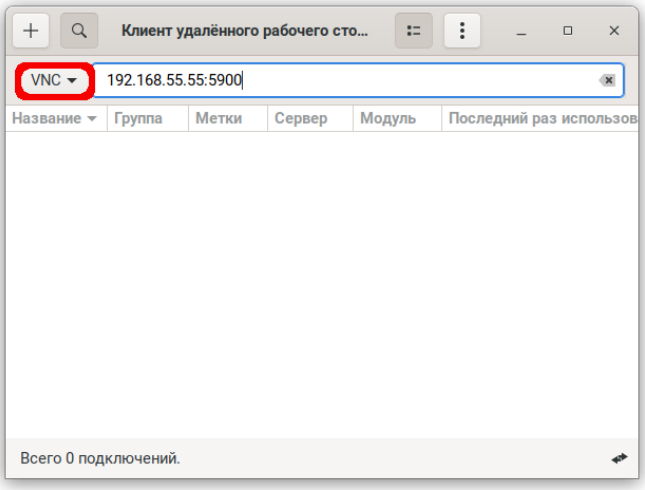
Установите приложение для удалённого подключения «Remmina», выполнив в «Терминале» следующую команду:

```
$ sudo dnf install remmina
```

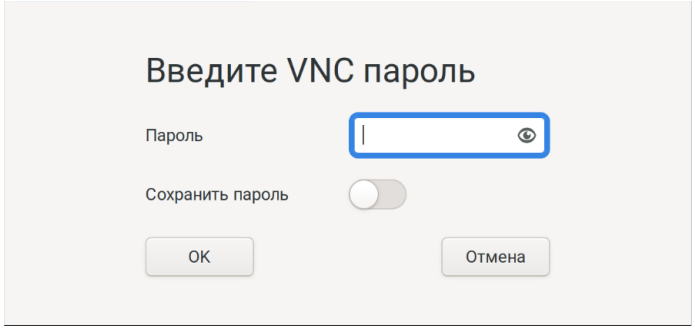
В главном меню появится ярлык приложения «Remmina»



Откройте приложение, поменяйте протокол подключения на VNC, укажите IP-адрес и порт, затем нажмите «Ввод» на клавиатуре.



Для завершения подключения к устройству, укажите пароль.



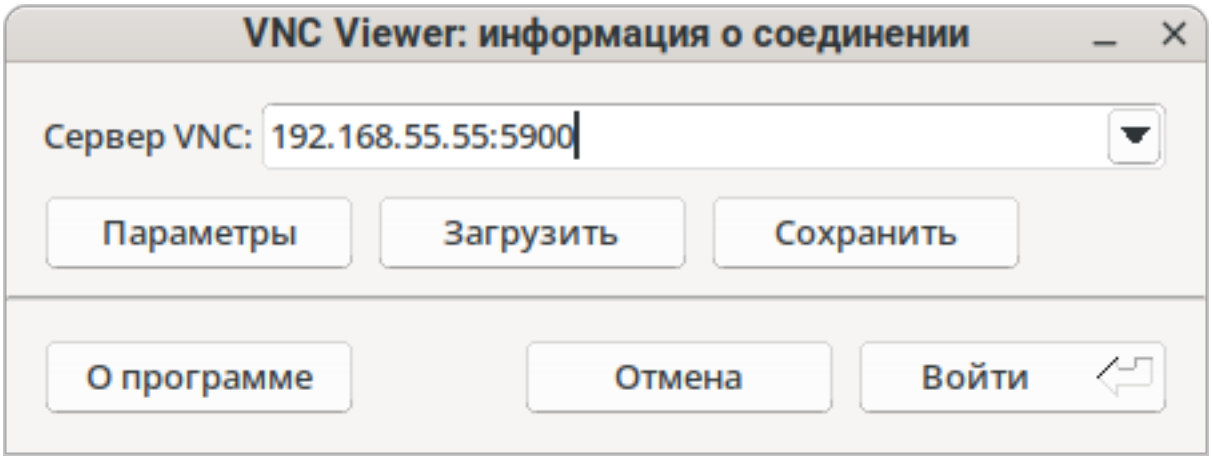
Удалённое подключение к ОС МСВСфера 9 с помощью TigerVNC

Установите приложение для удалённого подключения «TigerVNC», выполнив в «Терминале» следующую команду:

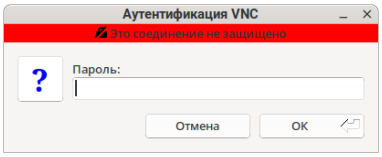
```
$ sudo dnf install tigervnc
```

В главном меню появится ярлык приложения «TigerVNC»:

Откройте приложение и укажите IP-адрес и порт, затем нажмите «Ввод» на клавиатуре.



Для завершения подключения к устройству, укажите пароль.



Централизованная аутентификация и авторизация пользователей

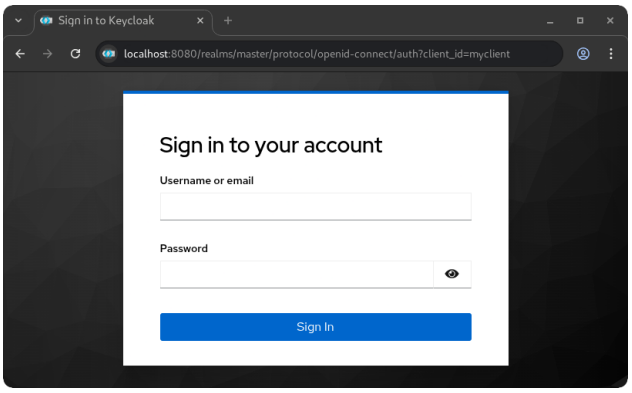
Введение

Для организации централизованной аутентификации и авторизации пользователей в ОС МСВСфера используется приложение Keycloak.

Ключевые функции Keycloak:

- **Единый вход (Single Sign-On — SSO)**

  Пользователи входят в систему один раз и получают доступ ко всем подключенным приложениям, не вводя логин и пароль для каждого из них. То есть приложения не должны реализовывать формы входа, проверять пользователей и хранить их данные. После входа в Keycloak пользователи могут переключаться между приложениями без повторной авторизации. Keycloak также обеспечивает единый выход (single sign-out), то есть пользователь выходит из всех приложений, использующих Keycloak, всего одним действием.



- **Аутентификация и авторизация**

  Обрабатывает входы пользователей, проверяет их подлинность и выдаёт токены для доступа к защищённым ресурсам.

- **Управление пользователями и ролями**

  Централизованно создает, управляет пользователями, их ролями и группами, определяя их права доступа к приложениям и данным. Если role-based модели авторизации недостаточно, Keycloak предоставляет возможность более тонкой настройки прав доступа, позволяющей управлять разрешениями для всех сервисов из административной консоли и задавать более специализированные политики безопасности.

- **Федерация пользователей**

  Keycloak имеет встроенную поддержку подключения к существующим серверам LDAP и Active Directory. Вы также можете создать собственный провайдер, если у вас пользователи хранятся в других системах, например, в реляционных базах данных.

- **Поддержка стандартных протоколов**

  Реализует стандарты аутентификации, такие как OpenID Connect, OAuth 2.0 и SAML.

- **Вход через социальные сети**

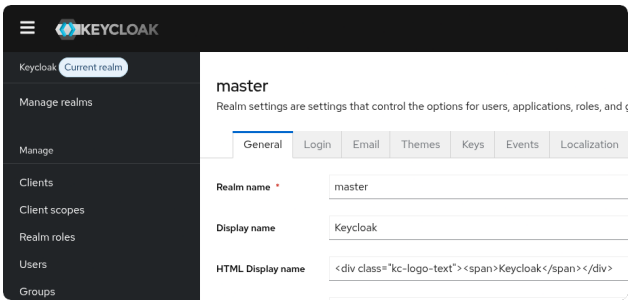
  Позволяет пользователям входить в систему, используя свои учётные записи в социальных сетях.

- **Гибкая интеграция**

  Благодаря REST API и поддержке различных фреймворков, Keycloak легко интегрируется с любыми фронтенд- и бэкенд-системами.

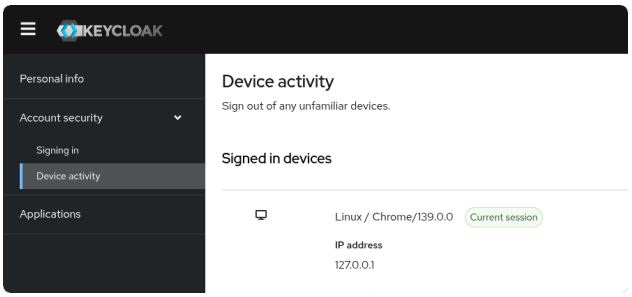
- **Административная панель управления**

  Предоставляет администраторам возможность централизованного управления всеми аспектами сервера Keycloak. Администратор может включать и отключать различные функции, настраивать провайдеров удостоверений и федерацию пользователей. Также есть возможность создавать и управлять приложениями и сервисами и задавать детализированные политики авторизации. Администраторы управляют пользователями, включая права доступа и сессии.



- **Панель управления аккаунтом**

Через панель управления аккаунтом пользователи могут самостоятельно настраивать свои учётные записи, изменять профиль, менять пароль и настраивать двухфакторную аутентификацию. Пользователи могут управлять своими сессиями и просматривать историю действий. Если включён вход через социальные сети или федерацию пользователей, пользователи могут привязывать аккаунты различных провайдеров, позволяя входить в один и тот же аккаунт через разные идентификаторы.



Установка

Перед началом работы с Keycloak на физическом или виртуальном сервере убедитесь, что ваше устройство имеет достаточный объём памяти и процессорных ресурсов для использования Keycloak.

Установка Keycloak

Для установки Keycloak из репозиториев ОС МСВСфера выполните следующую команду:

```
$ sudo dnf install keycloak
```

Запуск Keycloak

Для запуска Keycloak выполните следующую команду:

```
$  sudo /usr/lib/keycloak/bin/kc.sh start-dev
```

Опция `start-dev` запускает Keycloak в режиме разработки. В этом режиме вы можете быстро познакомиться с Keycloak и запустить его с настройками по умолчанию, удобными для разработчиков, например, для создания новой темы Keycloak.

Начало работы в Keycloak

Создание администратора

По умолчанию в Keycloak нет пользователя с правами администратора. Перед запуском нужно создать такого пользователя.

1. Откройте адрес `http://localhost:8080/`.
2. Заполните форму, указав желаемое имя пользователя и пароль.

Вход в панель управления администратора

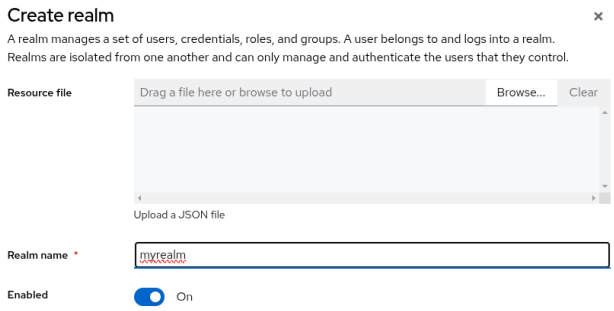
1. Перейдите в панель управления администратора Keycloak по адресу `http://localhost:8080/admin`.
2. Войдите, используя ранее созданные имя пользователя и пароль.

Создание областей

В Keycloak область (realm) позволяет администратору создавать изолированные группы приложений и пользователей. В системе уже есть одна область с названием `master`, которая предназначена только для администрирования Keycloak, но не для управления приложениями.

Для создания первой области, выполните следующие шаги:

1. Откройте панель управления администратора Keycloak по адресу `http://localhost:8080/admin`.
2. Нажмите **Create Realm** рядом с пунктом **Current realm**.
3. Укажите имя области в поле **Realm name**, например, `myrealm`.
4. Нажмите **Create**.

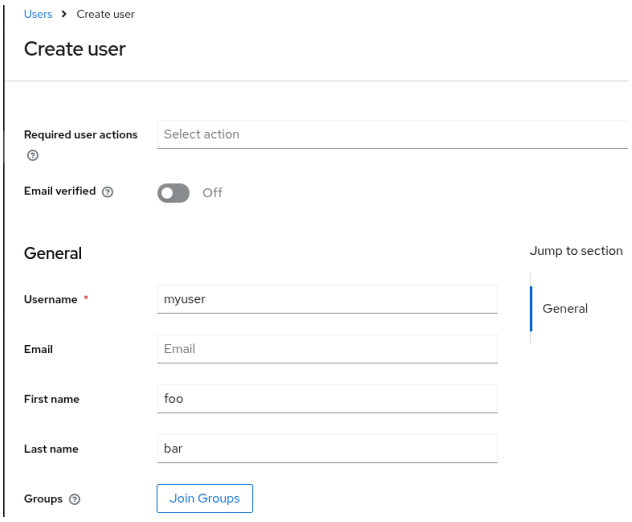


Создание пользователя

По умолчанию в области нет пользователей. Для создания пользователя выполните следующие шаги:

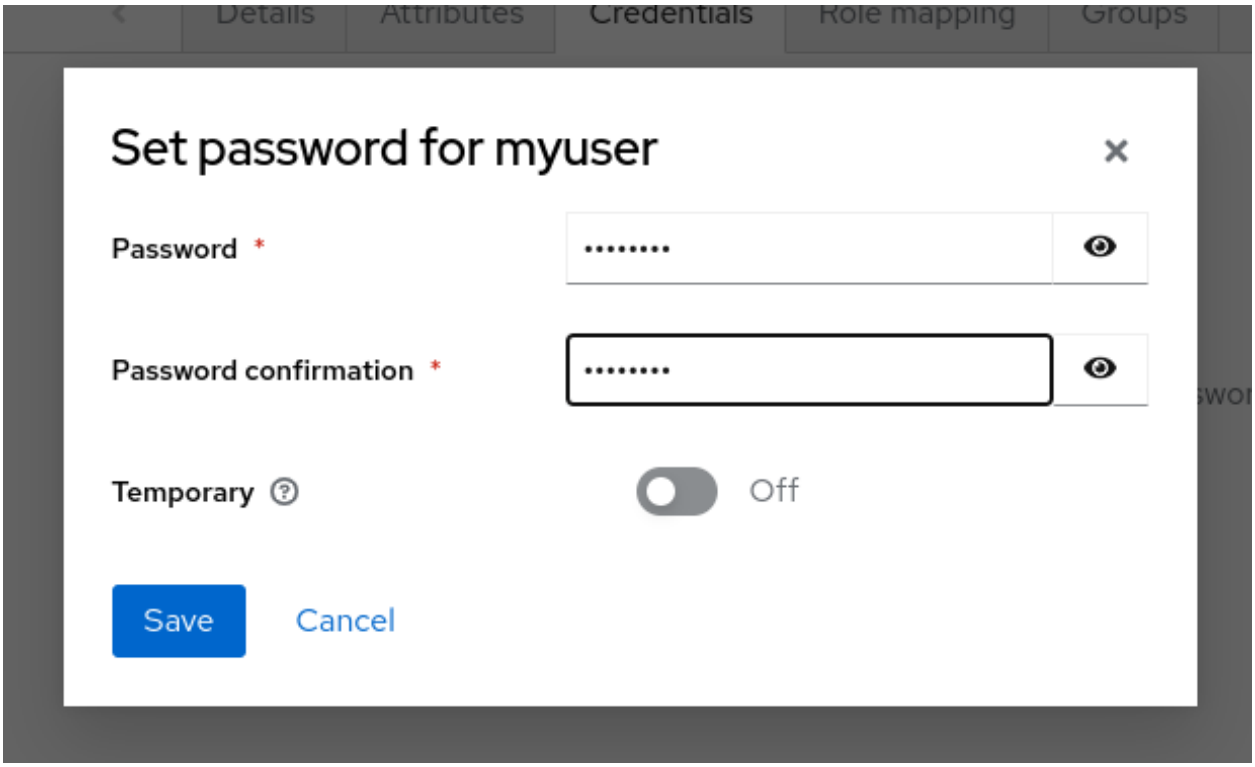
1. Убедитесь, что вы находитесь в нужной области (имя области отображаемом рядом с **Current realm**. В нашем примере это **myrealm**).
2. В левом меню выберите **Users**.
3. Нажмите **Create new user**.

4. Заполните форму, указав следующие данные:
   - **Username**: имя пользователя, например, `myuser`;
   - **First name**: имя пользователя;
   - **Last name**: фамилия пользователя;
5. Нажмите **Create**.



Чтобы пользователь мог войти в систему, ему необходимо установить пароль. Для установки пароля выполните следующие действия:

1. Перейдите на вкладку **Credentials** вверху страницы.
2. В форме **Set password** введите желаемый пароль.
3. Чтобы пароль не нужно было менять при первом входе, переведите слайдер **Temporary** в положение **Off**.

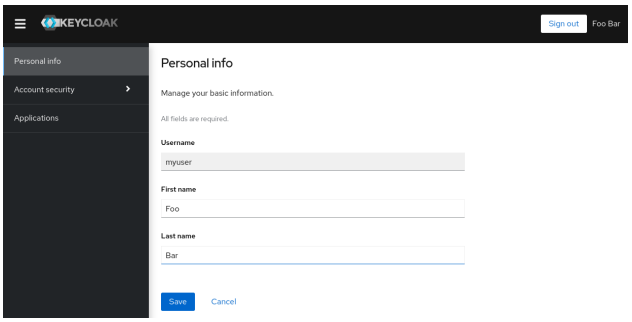


Вход в панель управления аккаунтом (Account Console)

Теперь вы можете войти в панель управления аккаунтом, чтобы проверить корректность настройки пользователя.

1. Откройте `http://localhost:8080/realms/myrealm/account`.
2. Войдите под своим пользователем (в нашем примере это `myuser`) и паролем, который вы задали ранее.

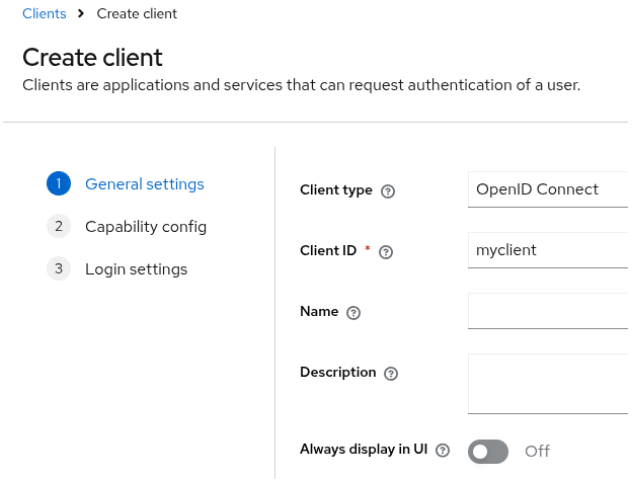
В панели управления аккаунтом пользователь может управлять своей учётной записью: менять профиль, настраивать двухфакторную аутентификацию, подключать аккаунты сторонних провайдеров идентификации.



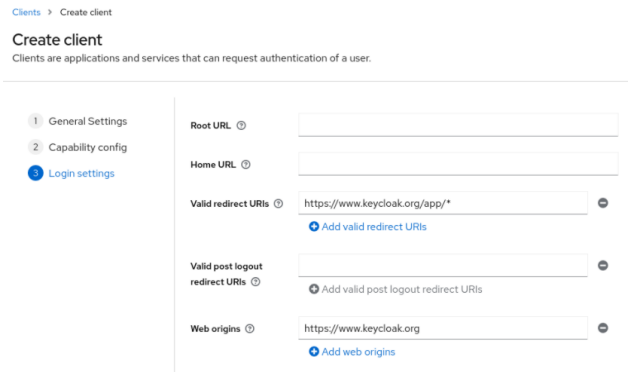
Защита первого приложения

Для защиты первого приложения сначала зарегистрируйте его в Keycloak:

1. Откройте `http://localhost:8080/admin`.
2. Убедитесь, что в поле **Current realm** выбрана ваша область (в нашем примере это `myrealm`).
3. Перейдите в раздел **Clients**.
4. Нажмите **Create client**.
5. Заполните форму:
   - **Client type**: например, `OpenID Connect`;
   - **Client ID**: например, `myclient`;



6. Нажмите **Next**.
7. Проверьте, что **Standard flow** включён.
8. Нажмите **Next**.
9. В разделе **Login settings** выполните следующие настройки:
   - В поле **Valid redirect URIs** введите `https://www.keycloak.org/app/*`.
   - В поле **Web origins** введите `https://www.keycloak.org`.
10. Нажмите **Save**.



Чтобы убедиться, что клиент успешно создан, можно воспользоваться тестовым SPA-приложением на сайте Keycloak:

1. Перейдите по адресу https://www.keycloak.org/app/.
2. Нажмите **Save** для использования настроек по умолчанию.
3. Нажмите **Sign in** и авторизуйтесь в приложении через запущенный ранее сервер Keycloak.

Конфигурация Keycloak

Keycloak загружает настройки из четырёх источников, перечисленных здесь в порядке приоритета применения.

1. Параметры командной строки.
2. Переменные окружения.
3. Параметры, определённые в файле `/usr/lib/keycloak/conf/keycloak.conf` или в пользовательском конфигурационном файле.
4. Конфиденциальные параметры, определённые в пользовательском файле Java KeyStore.

Если параметр задан в нескольких источниках, приоритет имеет тот источник, который стоит выше в списке. Например, значение, заданное в командной строке, будет иметь больший приоритет, чем переменная окружения для того же параметра.

Пример: настройка параметра db-url-host

Таблица 1: Пример того, как значение `db-url` может быть задано из четырёх источников конфигурации

Источник	Формат
Параметры командной строки	`--db-url=cliValue`
Переменная окружения	`--KC_DB_URL=envVarValue`
Конфигурационный файл	`db-url=confFileValue`
Файл Java KeyStore	`kc.db-url=keystoreValue`

Поскольку приоритет выше у командной строки, при запуске будет использовано значение `cliValue`.

Если параметр командной строки не задан, то будет использовано значение из переменной окружения `envVarValue`. Если и там нет значения, берётся конфигурация из файла, и наконец из файла KeyStore, который имеет самый низкий приоритет.

Форматы конфигурации

В конфигурации используется *единый для каждого источника* формат, что упрощает конвертацию ключ/значение между источниками. Обратите внимание, что такие форматы применимы и к параметрам spi.

Форматы для каждого источника:

- Параметры командной строки: `--<ключ-с-дефисами>=<значение>`. Для некоторых параметров существует сокращённая запись `-<аббревиатура>=<значение>`.
- Переменные окружения: `KC_<ключ_с_подчёркиваниями>=<значение>`, где ключ — в верхнем регистре.
- Конфигурационный файл: `<ключ-с-дефисами>=<значение>`
- Java KeyStore: `kc.<ключ-с-дефисами>`, где `<значение>` — это пароль, хранящийся в KeyStore.

Форматы параметров командной строки

Keycloak поддерживает множество параметров командной строки для настройки. Чтобы посмотреть их список, выполните команду:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start --help
```

Формат использования переменных окружения в файле конфигурации

В файле `keycloak.conf` можно использовать подстановки значений из переменных окружения, используя синтаксис `${ENV_VAR}`:

```
db-url-host=${MY_DB_HOST}
```

Если переменная окружения не определена, можно указать значение по умолчанию через двоеточие `:`:

```
db-url-host=${MY_DB_HOST:mydb}
```

Формат указания конкретного файла конфигурации

По умолчанию сервер читает настройки из `/usr/lib/keycloak/conf/keycloak.conf`. Для новой установки в этом файле только закомментированные параметры с подсказками.

Чтобы использовать другой файл, укажите его через опцию `--config-file`:

```
$ sudo /usr/lib/keycloak/bin/kc.sh --config-file=/path/to/config.conf start
```

Настройка конфиденциальных параметров через Java KeyStore

С помощью KeyStore Configuration Source можно загружать параметры напрямую из Java KeyStore файла, используя опции:

- `--config-keystore` — путь к файлу KeyStore.
- `--config-keystore-password` — пароль от KeyStore.
- `--config-keystore-type` — тип KeyStore (по умолчанию `PKCS12`).

Пароли в KeyStore должны иметь алгоритм шифрования с паролем (`PBE`), где ключ выводится из пароля KeyStore.

Создать такой KeyStore можно с помощью команды `keytool`:

```
keytool -importpass -alias kc.db-password -keystore keystore.p12 -storepass keystorepass -
↪storetype PKCS12 -v
```

Пароль — это значение для свойства `kc.db-password`.

Запустите сервер со следующими параметрами:

```
sudo /usr/lib/keycloak/bin/kc.sh start --config-keystore=/путь/к/keystore.p12 --config-keystore-
↪password=keystorepass --config-keystore-type=PKCS12
```

Формат для встроенных параметров Quarkus

Обычно, параметров Keycloak достаточно для настройки сервера, но если нужен дополнительный функционал, не предусмотренный в Keycloak, можно использовать параметры из фреймворка Quarkus.

Использование таких параметров не рекомендуется, так как они не поддерживаются Keycloak напрямую.

Для использования параметров Quarkus:

1. Создайте файл `quarkus.properties` в папке `conf`.
2. Определите нужные параметры в этом файле.

Можно установить только подмножество параметров, доступных в Quarkus (Документация Quarkus).

Обратите внимание:

- Значок замка в документации Quarkus означает параметр, применяемый на этапе сборки (build time).
- Отсутствие замка — runtime-свойство, применяемое во время работы.

Некоторые параметры Quarkus, например `quarkus.http.port`, уже используются в Keycloak и переопределяются настройками Keycloak.

Использование спецсимволов в значениях

Keycloak и Quarkus поддерживают выражения вида `${ключ}` для подстановки значений.

Чтобы отключить интерпретацию выражения, используйте символ `\`. Особенно это касается символа `$`, который надо экранировать при необходимости.

Например, для значения `my$$password` укажите `my\$\$password`.

Обратите внимание, что `\` нужно экранировать или использовать кавычки. Например:

- bash с одинарными кавычками: `--db-password='my\$\$password'`
- bash с двойными кавычками: `--db-password="my\\$\\$password"`
- В файлах свойств: `kc.db-password=my\\$\\$password`

Форматы ключей переменных окружения с особыми символами

Неалфавитно-цифровые символы в ключах замещаются на `_` в переменных окружения.

При трансформации обратно из переменной в ключ параметра `_` заменяется на `-`, кроме случаев для логирования, где `_` заменяется на `.` (классы и пакеты содержат точки).

Предупреждение

Автоматическая конвертация может нарушать ожидаемый ключ. Например, `kc.log-level-package.class_name` становится переменной `KC_LOG_LEVEL_PACKAGE_CLASS_NAME`, которая при обратном преобразовании превращается в `kc.log-level-package.class.name`, что может не совпадать с ожидаемым ключом.

Варианты решения:

- Создайте запись в `keycloak.conf` со ссылкой на переменную окружения, например: `kc.log-level-package.class_name=${CLASS_NAME_LEVEL}`

- Или используйте пару переменных окружения: `KC_UNIQUEIFIER=значение` и `KCKEY_UNIQUEIFIER=ключ` Например: `KC_MYKEY=debug` и `KCKEY_MYKEY=log-level-package.class_name`

Запуск Keycloak

Keycloak можно запускать в режиме разработки (development) и в режиме эксплуатации (production). Каждый режим устанавливает разные настройки по умолчанию исходя из целей.

Запуск Keycloak в режиме разработки (development)

Режим разработки подходит для быстрого знакомства с Keycloak или при работе над темой.

Запуск:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start-dev
```

Настройки по умолчанию в режиме разработки:

- HTTP включён:
- Строгая проверка имени хоста отключена;
- Используется локальный кэш (не распределённый);
- Кэширование тем и шаблонов отключено.

Запуск Keycloak в режиме эксплуатации (production)

Режим эксплуатации используется для развертывания в рабочих средах с принципом *безопасности по умолчанию*.

Запуск:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start
```

Без дополнительной настройки команда выдаст ошибку — это сделано специально, чтобы предотвратить запуск без обязательной настройки имени хоста и HTTPS/TLS.

Настройки по умолчанию в режиме эксплуатации:

- HTTP отключён (необходим HTTPS);
- Ожидается конфигурация имени хоста;
- Ожидается настройка HTTPS/TLS.

В файле `/usr/lib/keycloak/conf/keycloak.conf` по умолчанию содержатся закомментированные примеры параметров для запуска системы, готовой к эксплуатации (production-ready).

Создание начального пользователя-администратора

Начального администратора можно создать через веб-интерфейс при подключении с локального адреса (localhost), либо задать через переменные окружения:

- `KC_BOOTSTRAP_ADMIN_USERNAME=<имя_администратора>`;
- `KC_BOOTSTRAP_ADMIN_PASSWORD=<пароль_администратора>`.

Keycloak при первом запуске создаст такого пользователя.

Если администратор уже существует, а переменные окружения заданы, при запуске появится сообщение об ошибке создания, но сервер запустится.

Дополнительных пользователей можно создавать в панели управления администратора или с помощью утилиты командной строки `/usr/lib/keycloak/bin/kcadm.sh` (запуск через `sudo`).

Использование системных переменных в конфигурации областей (realms)

Администраторы областей могут ссылаться на системные переменные (переменные окружения и свойства системы) в настройках областей и его компонентов.

По умолчанию использование системных переменных запрещено, за исключением тех, что явно разрешены через параметр конфигурации:

```
spi-admin--allowed-system-variables
```

Этот параметр принимает перечень ключей через запятую, которые можно подставлять из системных переменных.

Пример запуска с разрешением переменных `FOO` и `BAR`:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start --spi-admin--allowed-system-variables=FOO,BAR
```

Настройка TLS

Необходимо настроить HTTPS-сертификаты Keycloak для входящих и исходящих запросов.

Transport Layer Security (TLS) — это важный протокол, обеспечивающий обмен данными по защищённому каналу. Для рабочих сред никогда не используйте HTTP для доступа к Keycloak, он обрабатывает конфиденциальные данные при взаимодействии с другими приложениями.

Keycloak можно настроить для загрузки сертификатов и ключей из файлов в формате PEM или из Java Keystore. Если заданы оба варианта, приоритет отдаётся PEM-файлам.

Предоставление сертификатов в формате PEM

Если у вас есть пара соответствующих файлов сертификата и приватного ключа в формате PEM, настройте Keycloak для их использования, запустив следующую команду:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start --https-certificate-file=/path/to/certificate.pem --
↪https-certificate-key-file=/path/to/key.pem
```

Keycloak создаст из этих файлов хранилище ключей в памяти и будет использовать его.

Предоставление Keystore

Если файл хранилища ключей не настроен явно, но параметр `http-enabled` выключен, Keycloak ищет файл `/usr/lib/keycloak/conf/server.keystore`.

Также можно использовать существующее хранилище ключей, запустив команду:

```
/usr/lib/keycloak/bin/kc.sh start --https-key-store-file=/path/to/existing-keystore
```

Поддерживаемые расширения файлов хранилища ключей:

- `.p12`, `.pkcs12`, `.pfx` — файлы в формате pkcs12.
- `.jks`, `.keystore` — файлы в формате JKS.
- `.key`, `.crt`, `.pem` — файлы в формате PEM.

Если ваше хранилище не имеет расширения, соответствующего типу файла, необходимо дополнительно указать опцию `https-key-store-type`.

Установка пароля для Keystore

Вы можете задать надёжный пароль для хранилища при помощи опции `https-key-store-password`:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start --https-key-store-password=<password>
```

Если пароль не указан, будет использоваться пароль по умолчанию — `password`.

Защита учётных данных

Во избежание указания пароля в открытом виде в командной строке или в файле `/usr/lib/keycloak/conf/keycloak.conf` рекомендуется применять лучшие практики безопасности: использовать хранилища (vault) или примонтированные секреты.

Перезагрузка сертификатов и ключей

По умолчанию Keycloak перезагружает сертификаты, ключи и хранилища, указанные в опциях `https-*`, каждый час. Это полезно в средах с частой ротацией ключей, и позволяет обновлять их без перезапуска сервера.

Вы можете изменить интервал перезагрузки с помощью опции `https-certificates-reload-period`. Значение может быть указано как `java.time.Duration`, целое число секунд или число с суффиксом времени [`ms`, `h`, `m`, `s`, `d`]. Интервал должен быть больше 30 секунд. Значение `-1` отключит перезагрузку.

Настройка доверенных сертификатов

Настройте хранилище доверенных сертификатов (Truststore) Keycloak для работы через TLS.

Когда Keycloak взаимодействует с внешними сервисами или принимает входящие соединения по TLS, он должен проверять удалённый сертификат, чтобы убедиться, что соединяется с доверенным сервером. Это необходимо для предотвращения атак типа «man-in-the-middle» («человек посередине»).

Сертификаты клиентов или серверов, а также сертификаты центра сертификации (CA), выдавшего эти сертификаты, должны быть добавлены в хранилище доверенных сертификатов. Это хранилище затем нужно настроить для использования Keycloak.

Настройка системного хранилища доверенных сертификатов

Сертификаты, включённые в стандартное системное хранилище сертификатов Java, всегда считаются доверенными. Если вам нужны дополнительные сертификаты — что актуально, например, при использовании самоподписанных сертификатов или внутренних центров сертификации, не распознаваемых JRE, — их можно положить в директорию `conf/truststores` или её поддиректории. Сертификаты могут быть в формате PEM или в файлах PKCS12 с расширениями `.p12`, `.pfx` или `.pkcs12`. Если используется PKCS12, сертификаты должны быть без шифрования, то есть без пароля.

Если необходим альтернативный путь, воспользуйтесь опцией `--truststore-paths`, чтобы указать дополнительные файлы или директории с PEM- или PKCS12-файлами. Пути указываются относительно директории запуска Keycloak, поэтому рекомендуется использовать абсолютные пути. Если указан каталог, он будет просканирован рекурсивно на предмет файлов сертификатов.

После добавления всех нужных сертификатов, хранилище будет использоваться в качестве системного по умолчанию через свойства `javax.net.ssl`, а также по умолчанию внутри самого Keycloak.

Пример запуска:

```
$ sudo /usr/lib/keycloak/bin/kc.sh start --truststore-paths=/opt/truststore/myTrustStore.pfx,/
↪opt/other-truststore/myOtherTrustStore.pem
```

Также можно напрямую задать собственные системные свойства `javax.net.ssl` для хранилища, но рекомендуется использовать именно `--truststore-paths`.

Политика проверки имени хоста

Вы можете уточнить, как осуществляется проверка имени хоста при TLS-соединениях, с помощью свойства `tls-hostname-verifier`.

- `DEFAULT` (значение по умолчанию) разрешает использовать подстановочные символы (wildcards) в поддоменах (например, `\*.foo.com`), которые соответствуют именам с таким же числом уровней (например, `a.foo.com`, но не `a.b.foo.com`). При этом применяются правила и исключения для публичных суффиксов на основе https://publicsuffix.org/list/.
- `ANY` — имя хоста не проверяется. Этот режим нельзя использовать в рабочей среде.
- `WILDCARD` (устаревший) разрешает подстановочные символы в поддоменах (например, `\*.foo.com`) и может сопоставлять имена с любым количеством уровней (например, `a.b.foo.com`). Рекомендуется использовать вместо него `DEFAULT`.
- `STRICT` (устаревший) разрешает подстановочные символы в поддоменах (например, `\*.foo.com`) и сопоставляет имена с таким же числом уровней (например, `a.foo.com`, но не `a.b.foo.com`) с некоторыми ограничениями. Используйте вместо него `DEFAULT`.

Обратите внимание, что это свойство не применяется к защищённым LDAP-соединениям, которые требуют строгой проверки имени хоста.

Настройка базы данных

Настройте реляционную базу данных для Keycloak для хранения данных пользователей, клиентов и областей.

В этом руководстве объясняется, как настроить сервер Keycloak для хранения данных в реляционной базе данных.

Поддерживаемые базы данных

Сервер поддерживает различные базы данных без дополнительной настройки. Вы можете узнать доступные базы данных, просмотрев ожидаемые значения для параметра конфигурации `db`. В таблице перечислены поддерживаемые базы данных и протестированные версии.

Таблица 2: Поддерживаемые базы данных и протестированные версии

База данных	Значение параметра	Протестированная версия
MariaDB Server	`mariadb`	11.4
Microsoft SQL Server	`mysql`	2022
MySQL	`mysql`	8.4
Oracle Database	`oracle`	23.5
PostgreSQL	`postgres`	17
Amazon Aurora PostgreSQL	`postgres`	16.8

По умолчанию сервер использует базу данных `dev-file`. Это база данных по умолчанию, которую сервер использует для сохранения данных и которая существует только для разработки. База данных `dev-file` не подходит для использования в рабочей среде, и перед развёртыванием её необходимо заменить.

Настройка базы данных

Для каждой поддерживаемой базы данных сервер предлагает рекомендуемые значения по умолчанию для упрощения настройки. Вам нужно лишь указать основные параметры, например, хост базы данных и учётные данные.

Конфигурацию можно задать как при выполнении команды `build`, так и при запуске сервера (`start`):

1. Использование команды `build`, а затем оптимизированного `start` (рекомендуется).

   Сначала укажите минимальные параметры подключения к базе в файле `/usr/lib/keycloak/conf/keycloak.conf`:

   ```
   # Поставщик базы данных.
   db=postgres

   # Имя пользователя базы данных.
   db-username=keycloak

   # Пароль пользователя базы данных.
   db-password=change_me

   # Хост JDBC URL выбранного поставщика
   db-url-host=keycloak-postgres
   ```

   Затем выполните команды для сборки нового оптимизированного образа сервера и его запуска:

   ```
   $ sudo /usr/lib/keycloak/bin/kc.sh build

   $ sudo /usr/lib/keycloak/bin/kc.sh start --optimized
   ```

2. Использование только команды `start` (без `--optimized`):

   ```
   /var/lib/keycloak/bin/kc.sh start --db postgres --db-url-host keycloak-postgres --db-
   ↪username keycloak --db-password change_me
   ```

Примечание

Примеры выше показывают минимальные параметры для подключения, но здесь раскрыт пароль базы, что нежелательно. Лучше использовать файл `/var/lib/keycloak/conf/keycloak.conf`, переменные окружения или keystore для хранения пароля.

По умолчанию используется схема `keycloak`, но вы можете изменить её с помощью параметра конфигурации `db-schema`.

```
/var/lib/keycloak/bin/kc.sh import --help
/var/lib/keycloak/bin/kc.sh export --help
/var/lib/keycloak/bin/kc.sh bootstrap-admin --help
```

Переопределение стандартных настроек подключения

Сервер использует JDBC для работы с базой данных. Если стандартных настроек недостаточно, можно указать полный JDBC URL через опцию `db-url`.

Пример команды для PostgreSQL:

```
/var/lib/keycloak/bin/kc.sh start --db postgres --db-url jdbc:postgresql://mypostgres/mydatabase
```

Обратите внимание, что в командной строке нужно экранировать специальные символы, например `;`. Чтобы избежать этого, лучше указать URL в конфигурационном файле.

Настройка поддержки Unicode в базе данных

Поддержка Unicode зависит от того, разрешает ли база данных использовать Unicode в полях VARCHAR и CHAR.

- Если это возможно, Unicode обычно поддерживается, но при этом может снижаться максимальная длина поля.
- Если поддерживается только Unicode в NVARCHAR и NCHAR, то все текстовые поля, построенные на VARCHAR и CHAR, скорее всего, не смогут правильно хранить все Unicode символы.

Схема базы предоставляет полную поддержку Unicode только для специальных полей:

- **Области (Realms)**: отображаемое имя, HTML-имя, локализации (ключи и значения);
- **Поставщики федерации (Federation Providers)**: отображаемое имя;
- **Пользователи (Users)**: имя пользователя, имя, фамилия, названия и значения атрибутов;
- **Группы (Groups)**: название, названия и значения атрибутов;

- **Роли (Roles)**: название;
- Описания объектов.

Во всех остальных полях символы ограничены набором кодировки базы, который часто 8-битный. В некоторых СУБД можно включить кодировку UTF-8 для полноценной поддержки Unicode в любых текстовых полях. Учтите, что при этом максимальная длина строк может уменьшиться по сравнению с 8-битным режимом.

Настройка Unicode в MySQL

MySQL поддерживает Unicode в VARCHAR и CHAR, если база создана с соответствующей кодировкой. Следует использовать команду `CREATE DATABASE` с кодировкой, например:

```
CREATE DATABASE mydb CHARACTER SET utf8;
```

Обратите внимание, что `utf8mb4` не поддерживается по причине иной организации хранения. При использовании utf8 ограничения по длине зависят от количества символов, а не байт. Если база не поддерживает Unicode по умолчанию, только специальные поля хранят Unicode.

Чтобы включить поддержку Unicode:

1. Запустите MySQL сервер.
2. В параметрах соединения JDBC добавьте: `characterEncoding=UTF-8`.

Настройка Unicode в PostgreSQL

Unicode поддерживается, если кодировка базы — UTF8. В этом случае Unicode доступен во всех полях без ограничения длины. Драйвер JDBC не требует дополнительных настроек.

Проверьте текущую кодировку:

```
show server_encoding;
```

Если не UTF8, создайте базу с нужной кодировкой:

```
create database keycloak with encoding 'UTF8';
```

Настройка оборудования

Управление принтерами

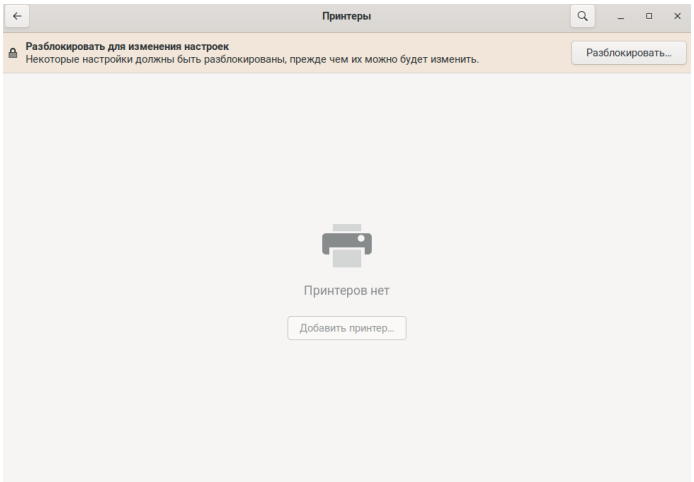
Введение

Настройка и управлении принтерами в графическом интерфейсе ОС МСВСфера производится в приложении «**Настройки**». Перейти в «**Настройки**» вы можете из главного меню, набрав в строке поиска «настройки» и нажав на приложение правой кнопкой мыши, или нажав значок «**Шестерёнки**» в системной панели или в главном меню. Перед добавлением и настройкой принтера убедитесь, что он подключён к сети питания и включён.

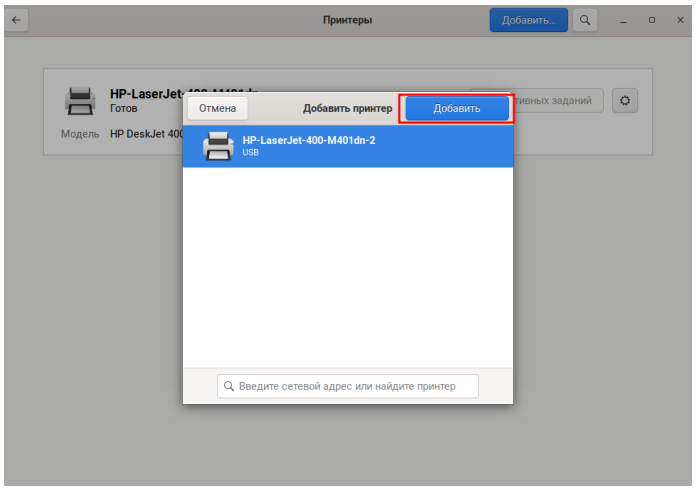
Добавление принтера в систему

Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Принтеры**».

Для любой учётной записи (кроме суперпользователя) при начальном входе некоторые настройки будут заблокированы. Для разблокировки необходимо пройти аутентификацию.



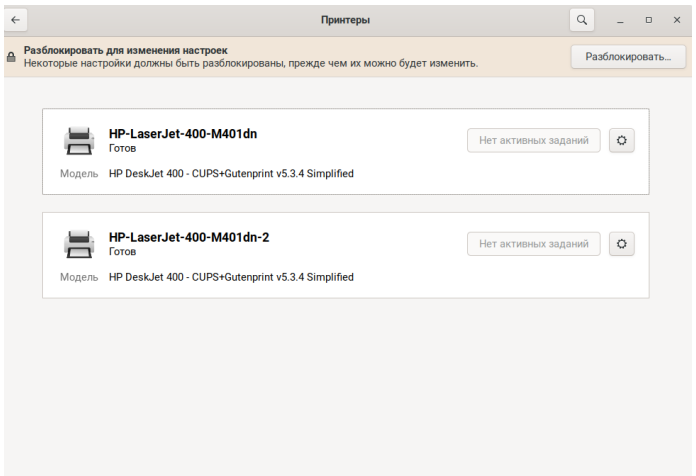
После успешного прохождения аутентификации активируется кнопка «**Добавить принтер**». Нажмите на неё, откроется окно «**Добавить принтер**».



Примечание

Обычно при подключении принтера, он добавляется в систему автоматически и сразу будет виден в разделе «Настройки» → «Принтеры». При поиске принтера вы также можете указать его сетевой адрес вручную в строке поиска.

Выберите принтер и нажмите «**Добавить**». Принтер появится в списке.

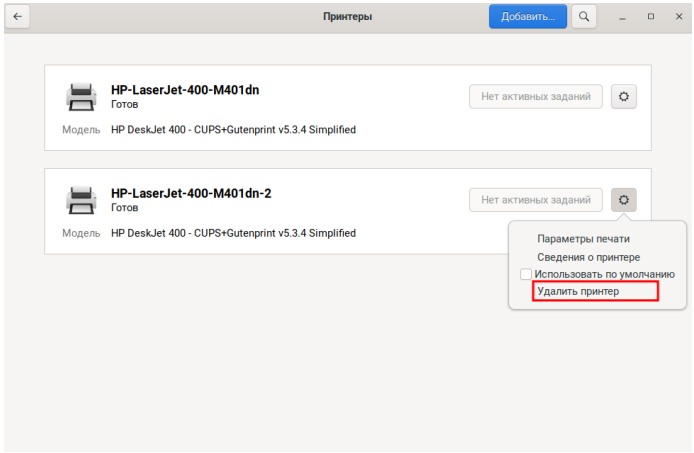


Удаление принтера

Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Принтеры**».

Для любой учётной записи (кроме суперпользователя) при начальном входе некоторые настройки будут заблокированы. Для разблокировки необходимо пройти аутентификацию.

Выберите необходимый принтер из списка и нажмите на значок «Шестерёнки», в открывшемся выпадающем списке нажмите на «Удалить принтер». Принтер будет удалён немедленно и перестанет отображаться в списке.

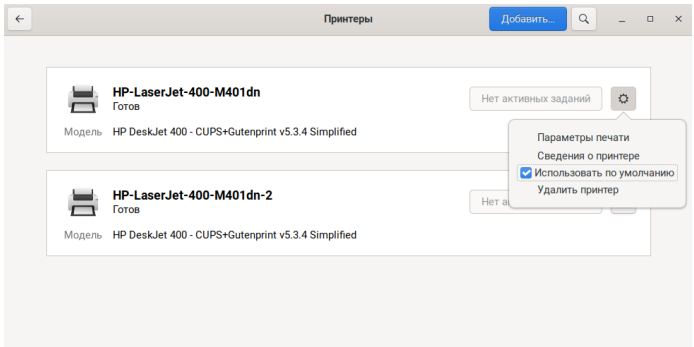


Изменение принтера по умолчанию

Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Принтеры**».

Для любой учётной записи (кроме суперпользователя) при начальном входе некоторые настройки будут заблокированы. Для разблокировки необходимо пройти аутентификацию.

Выберите необходимый принтер из списка и нажмите на значок «Шестерёнки», в открывшемся выпадающем списке поставьте галочку в пункте «Использовать по умолчанию».

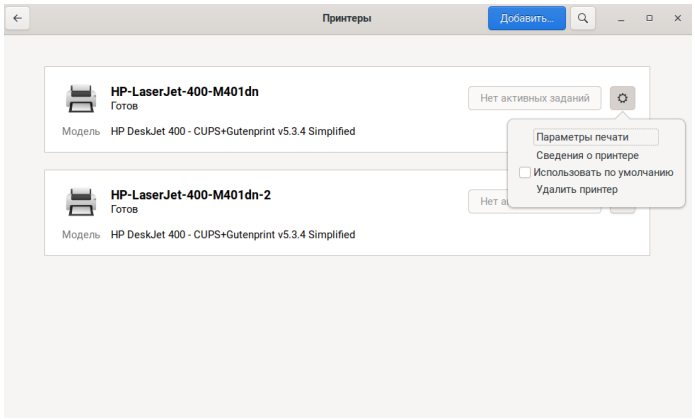


Настройка параметров печати

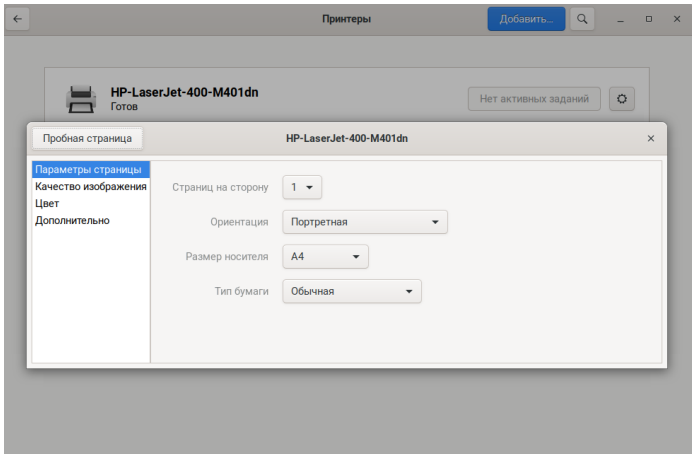
Откройте приложение «**Настройки**» любым удобным способом и перейдите в раздел «**Принтеры**».

Для любой учётной записи (кроме суперпользователя) при начальном входе некоторые настройки будут заблокированы. Для разблокировки необходимо пройти аутентификацию.

Выберите необходимый принтер из списка и нажмите на значок «Шестерёнки», в открывшемся выпадающем списке выберите «Параметры печати».



В открывшемся окне с названием принтера вы можете настроить все необходимые параметры.



Добавление водяных знаков (watermark) на документах при печати

Функция добавления водяных знаков на печатаемых документах в ОС МСВСфера реализована с помощью фильтра для системы печати cups. Фильтр работает таким образом, что в момент печати он перехватывает отправляемый на печать документ и добавляет в него фоном PDF-документ, содержащий водяные знаки. С помощью этого фильтра вы можете добавить водяные знаки на любой печатаемый документ — PDF-документ, изображение, текстовый или табличный документ любого формата. Функция также распространяется на печать из браузера.

PDF-документ, содержащий водяные знаки, находится в `/usr/share/cups/data/default-watermark.pdf`.

Вы также можете создать свой собственный PDF-документ с требуемыми водяными знаками и положить его по следующему адресу `/usr/share/cups/data/custom-watermark.pdf`. После этого водяные знаки будут браться уже из вашего PDF-документа.

Для начала работы с фильтром установите пакет `cups-filter-watermark` следующей командой (и при необходимости перезагрузите компьютер):

```
$ sudo dnf install cups-filter-watermark
```

Удалённое подключение USB-устройств по сети

Введение

USBIP — программное обеспечение, которое позволяет подключать по сети USB-устройства, физически подключённые к удалённому компьютеру, и использовать их так же, как если бы они были подключены к локальному компьютеру. Далее компьютер, к которому физически подключено USB-устройство, будет называться сервером, а тот, который будет использовать это устройство по сети — клиентом.

Обязательные условия

Для работы USBIP необходимо, чтобы клиент имел доступ к порту 3240 сервера. Пользователь, который настраивает USBIP на клиенте и на сервере, должен иметь административный доступ (`sudo`).

Установка и настройка ПО

Для установки USBIP и на сервере, и на клиенте нужно выполнить следующую команду:

```
$ sudo dnf install usbip kmod-usbip
```

Для установки графического интерфейса выполните следующую команду:

```
$ sudo dnf install usbip-gui
```

Настройка автоматического запуска служб

На сервере необходимо включить службу `usbip-server`:

```
$ sudo systemctl enable --now usbip-server
```

На клиенте — включить службу `usbip-client`:

```
$ sudo systemctl enable --now usbip-client
```

Настройка межсетевого экрана firewalld

Предупреждение

В приложении не реализованы функции аутентификации и авторизации при организации удалённого доступа к службе USBIP. Поэтому администратор должен настроить межсетевой экран таким образом, чтобы удалённый доступ был возможен только с авторизованных рабочих мест с фиксированными IP-адресами.

Настройка доступа при помощи межсетевого экрана firewalld

Для доступа USBIP удалённые компьютеры подключаются к порту `3240` по протоколу TCP. Например, чтобы открыть доступ к USBIP для компьютера, имеющего адрес `192.168.1.10`, нужно выполнить следующие команды:

```
$ sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.1.10
↪" port port="3240" protocol="tcp" accept'
```

При необходимости повторить выполнение команд для дополнительных адресов, после этого перезапустить `firewalld`:

```
$ sudo systemctl restart firewalld
```

Альтернативный способ

Создать новую зону, например, `allowedips`:

```
$ sudo firewall-cmd --permanent --new-zone=allowedips
```

Добавить доверенные IP-адреса в список источников (sources) этой зоны:

```
# Для 192.168.1.10
$ sudo firewall-cmd --permanent --zone=allowedips --add-source=192.168.1.10
```

```
# Для 192.168.1.20
$ sudo firewall-cmd --permanent --zone=allowedips --add-source=192.168.1.20
```

Открыть порт `3240/tcp` в созданной зоне:

```
$ sudo firewall-cmd --permanent --zone=allowedips –add-port=3240/tcp
```

Применить изменения:

```
$ sudo systemctl –reload firewalld
```

Настройка привязки устройства на сервере

Чтобы на стороне сервера сделать устройство доступным по сети, его нужно привязать к службе `usbipd`. Для этого нужно выполнить следующие действия. Предположим, нам нужно передать на клиентский компьютер USB-устройство «Aktiv Rutoken ECP». Сначала выясним идентификатор шины устройства. Для этого необходимо выполнить команду:

```
$ usbip list -l
- busid 1-1.2 (2357:0604)
  TP-Link : unknown product (2357:0604)

- busid 1-1.3 (17ef:60ee)
  Lenovo : unknown product (17ef:60ee)

- busid 1-1.4 (046d:c52b)
  Logitech, Inc. : Unifying Receiver (046d:c52b)

- busid 1-3 (0a89:0030)
  Aktiv : Rutoken ECP (0a89:0030)

- busid 1-7.1 (0d8c:0103)
  C-Media Electronics, Inc. : CM102-A+/102S+ Audio Controller (0d8c:0103)

- busid 1-7.2 (046d:082d)
  Logitech, Inc. : HD Pro Webcam C920 (046d:082d)
```

Из приведённого вывода команды видно, что значение `busid` (идентификатор шины) устройства `Rutoken ECP` равно `1-3`.

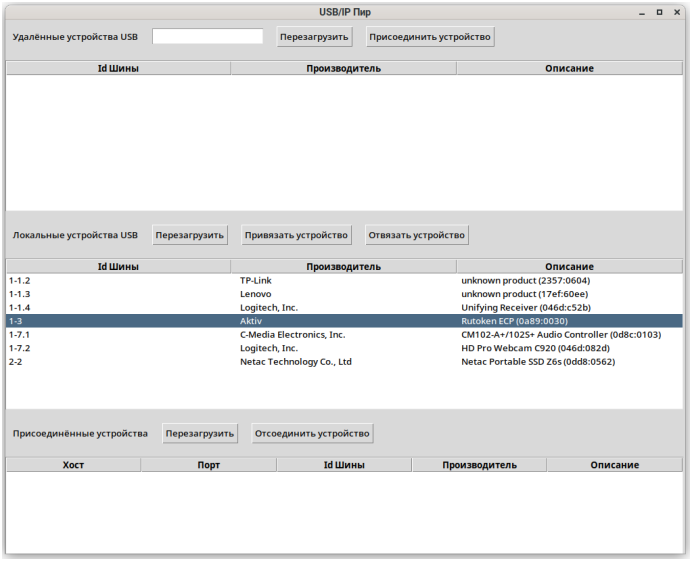
Далее необходимо выполнить привязку устройства:

```
$ sudo usbip bind -b 1-3
usbip: info: bind device on busid 1-3: complete
```

Для отключения устройства используется команда `unbind`:

```
$ sudo usbip unbind -b 1-3
usbip: info: unbind device on busid 1-3: complete
```

Эту же операцию можно выполнить при помощи графического интерфейса. Для этого запустите утилиту «Управление USBIP» из главного меню. В разделе «Локальные устройства USB» найдите нужное устройство, выделите его и нажмите кнопку «Привязать устройство». Для отключения устройства выделите устройство и нажмите кнопку «Отвязать устройство».



Подключение устройства к клиенту

Чтобы подключить устройство к клиенту, сначала нужно выполнить просмотр устройств, доступных на сервере. Для этого выполните команду:

```
$ sudo usbip list -r 192.168.10.62
Exportable USB devices
======================
- 192.168.10.62
      1-3: Aktiv : Rutoken ECP (0a89:0030)
         : /sys/devices/pci0000:00/0000:00:14.0/usb1/1-3
         : (Defined at Interface level) (00/00/00)
```

Где `192.168.10.62` — адрес сервера (можно указать также имя узла).

Таким образом видно, что устройство с `busid 1-3` привязано на сервере.

Подключение устройства:

```
$ sudo usbip attach -r 192.168.10.62 -b 1-3
```

Проверка подключения устройства:

```
$ lsusb | grep Rutoken
Bus 003 Device 004: ID 0a89:0030 Aktiv Rutoken ECP
```

Как видно, устройство `Rutoken` теперь подключено к клиентскому компьютеру.

Для отключения устройства сначала нужно определить, к какому порту виртуального хаба подключено устройство.

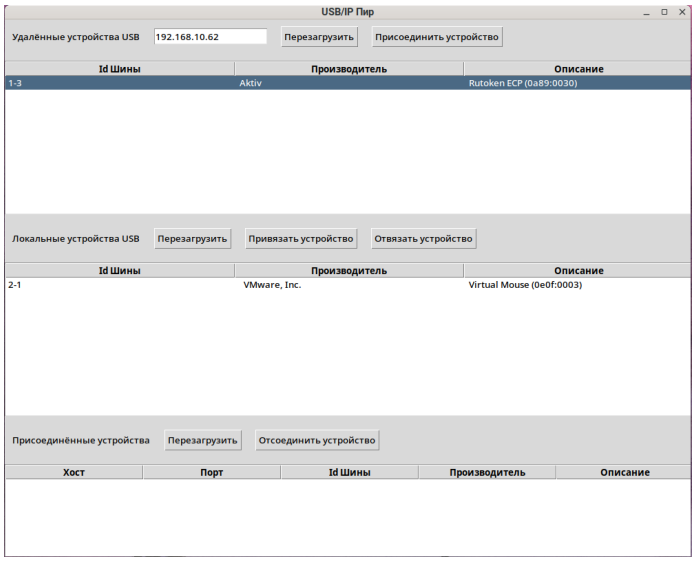
```
$ sudo usbip port
Imported USB devices
====================
Port 00: <Port in Use> at Full Speed(12Mbps)
     Aktiv : Rutoken ECP (0a89:0030)
     3-1 -> usbip://192.168.10.62:3240/1-3
         -> remote bus/dev 001/027
```

Из этого вывода видно, что устройство подключено к порту `0`. Теперь можно отключить его:

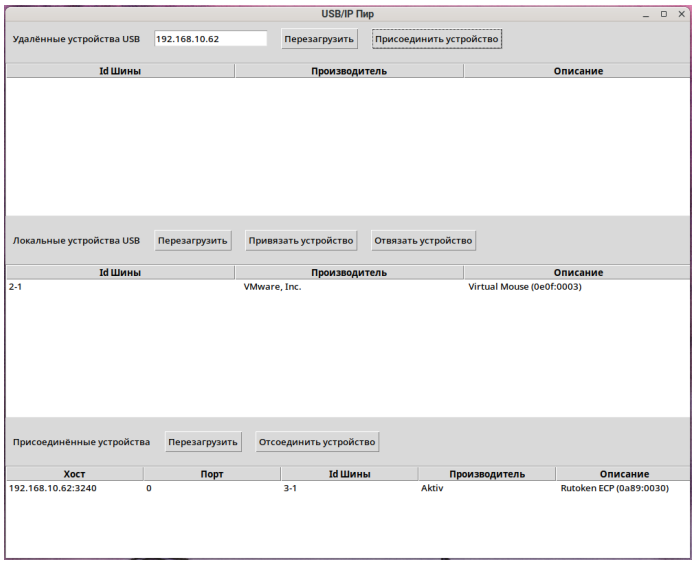
```
$ sudo usbip detach -p 0
usbip: info: Port 0 is now detached!
```

Подключение с использованием графического интерфейса

Для подключения устройства с помощью графического интерфейса запустите утилиту «Управление USBIP», введите адрес сервера в поле «Удалённые устройства USB» и нажмите кнопку «Перезагрузить». После этого устройство должно появиться в списке доступных. Для подключения устройства выделите его и нажмите кнопку «Присоединить устройство».



После этого устройство пропадает из списка доступных и появляется в списке присоединённых.



Для отключения выделите устройство в списке присоединённых и нажмите кнопку «Отсоединить устройство».

Управление пакетами

Введение и основные понятия

ОС МСВСфера представляет собой комплексную систему, которая обеспечивает стабильную и безопасную работу для пользователей.

Так как ОС МСВСфера собрана на базе ядра Linux, то в ней несколько приложений могут использовать одни и те же библиотеки или, например, одно приложение может использовать другое. С одной стороны это даёт возможность освободить место, занимаемое приложением, и снизить потребление ресурсов, а с другой стороны возникает необходимость обеспечения целостности системы.

Информация о всех необходимых приложению бинарных и конфигурационных файлах, о том, как их следует разместить в файловой системе, а также данные о зависимостях хранится в архиве специального формата, называемом **пакетом**.

В ОС МСВСфера форматом пакета является RPM (рекурсивный акроним RPM Package Manager, ранее Red Hat Package Manager), а сами файлы, содержащие пакеты, имеют расширение `.rpm`.

Как было упомянуто выше, приложения могут совместно использовать одни и те же библиотеки или даже целые программы, и здесь возникает понятие **зависимости**: в приложении может не хватать чего-то для работы, и ему для этого нужно другое приложение или библиотека. То есть один пакет начинает зависеть от другого. И удалив, например, одну библиотеку можно нарушить работу сразу нескольких приложений.

Для работы с пакетами и обеспечения целостности системы используются программы, называемые **пакетными менеджерами**. Они управляют пакетами: устанавливают, удаляют, обновляют, ведут учёт, выводят информацию, отслеживают версии и зависимости и пр.

В ОС МСВСфера пакетным менеджером является **DNF**.

Так как пакеты зависят друг от друга, то зачастую недостаточно установить только один пакет — нужно устанавливать сразу несколько, поэтому разработчики создают и поддерживают специальные централизованные серверы, называемые **репозиториями**, где хранятся различные пакеты. Пакетный менеджер видит зависимости каждого пакета, сам находит подходящие пакеты в репозитории и предлагает их установить.

Дистрибутив ОС МСВСфера имеет набор собственных репозиториев для всех поддерживаемых выпусков и архитектур, в которых содержится огромное количество приложений и программ.

Обычно некоторые пакеты, которые часто используют вместе, объединены в **группы**. Посмотреть список доступных групп поможет пакетный менеджер DNF.

Кроме групп также есть **модули**, которые тоже содержат сразу несколько пакетов, но при этом пакеты в модуле связаны версиями.

Пакетный менеджер DNF

Рассмотрим основные операции с пакетами, которые может выполнить пакетный менеджер DNF.

Найти нужный пакет

Для поиска пакета (даже не зная его точного имени) выполните следующую команду:

```
$ dnf search имя_пакета
```

В имени пакета вы можете использовать шаблоны, а также указывать только те буквы из названия, которые помните.

Пример: найдём пакет по первым буквам:

```
$ dnf search *fox
```

Результат работы команды:

```
$ dnf search *fox
==== Имя совпадение: *fox =================
firefox.x86_64 : Mozilla Firefox Web browser
```

Установить нужный пакет

Для установки пакета выполните следующую команду:

```
$ sudo dnf install имя_пакета
```

DNF проверит все зависимости и при обнаружении нужных, но ещё не установленных пакетов, установит их, пользуясь всеми доступными репозиториями.

Пример: установим пакет `firefox.x86_64`:

```
$ sudo dnf install firefox.x86_64
Зависимости разрешены.
==========================================================================
Пакет             Архитектура Версия                    Репозиторий Размер
==========================================================================
Установка:
 firefox           x86_64     128.11.0-1.el10_0.inferit appstream   125 M

Результат транзакции
==========================================================================
Установка  1 Пакет

Объем загрузки: 125 M
Объем изменений: 307 M
Продолжить? [д/Н]: д
Загрузка пакетов:
(1/1): firefox-128.11.0-1.el10_0.inferit.x86_64.rpm  7.7 MB/s | 125 MB     00:13
--------------------------------------------------------------------------
Общий размер                                         7.6 MB/s | 125 MB     00:14
Проверка транзакции
Проверка транзакции успешно завершена.
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
 Подготовка       :                                          1/1
 Установка        : firefox-128.11.0-1.el10_0.inferit.x86_64  1/1
 Запуск скриптлета: firefox-128.11.0-1.el10_0.inferit.x86_64   1/1

Установлен:
 firefox-128.11.0-1.el10_0.inferit.x86_64

Выполнено!
```

Обновить установленные пакеты

Для проверки наличия обновлений выполните команду `dnf check-upgrade`.

Пример работы команды для ОС МСВСфера 10:

```
$ dnf check-upgrade

MSVSphere 10 - AppStream          7.2 MB/s | 9.6 MB     00:01
MSVSphere 10 - BaseOS             4.2 MB/s | 3.6 MB     00:00
MSVSphere 10 - CRB                3.0 MB/s | 2.7 MB     00:00
MSVSphere 10 - Extras             1.8 MB/s | 989 kB     00:00

NetworkManager.x86_64            1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-adsl.x86_64       1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-bluetooth.x86_64  1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-libnm.x86_64      1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-team.x86_64       1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-tui.x86_64        1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-wifi.x86_64       1:1.42.2-6.el10_0.inferit  baseos
NetworkManager-wwan.x86_64       1:1.42.2-6.el10_0.inferit  baseos
```

Для обновления всей системы выполните следующую команду:

```
$ sudo dnf upgrade
```

Для обновления определённого пакета (и его зависимостей) выполните следующую команду:

```
$ sudo dnf upgrade имя_пакета
```

Удалить установленный пакет

Для удаления пакета выполните следующую команду:

```
$ dnf remove имя_пакета
```

Пример: удалим пакет `firefox.x86_64`:

```
$ sudo dnf remove firefox.x86_64
Зависимости разрешены.
================================================================================
 Пакет               Архитектура  Версия                     Репозиторий  Размер
================================================================================
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
Удаление:
firefox              x86_64       128.11.0-1.el10_0.inferit  @appstream   307 M

Результат транзакции
================================================================================
Удаление  1 Пакет

Освобожденное место: 307 M
Продолжить? [д/Н]: д
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
 Подготовка       :                                                    1/1
 Запуск скриптлета: firefox-128.11.0-1.el10_0.inferit.x86_64    1/1
 Удаление         : firefox-128.11.0-1.el10_0.inferit.x86_64    1/1
 Запуск скриптлета: firefox-128.11.0-1.el10_0.inferit.x86_64    1/1

Удален:
 firefox-128.11.0-1.el10_0.inferit.x86_64

Выполнено!
```

Вы можете увидеть, что также были удалены все пакеты, которые зависят от удаляемого.

Проверить целостность пакета

Для проверки целостности rpm-пакета выполните следующую команду:

```
$ rpm -V имя_rpm_пакета
```

В результате работы команды будет указана следующая информация:

- размер пакета
- полномочия
- тип
- владелец
- группа
- MD5-сумма
- дата последнего изменения пакета

Получить информацию об установленном пакете

Для получения подробной информации об установленном пакете выполните следующую команду:

```
$ dnf info имя_пакета
```

Пример работы команды для пакета `firefox.x86_64`:

```
$ dnf info firefox.x86_64
Установленные пакеты
Имя          : firefox
Версия       : 128.11.0
Выпуск       : 1.el10_0.inferit
Архитектура  : x86_64
Размер       : 307 M
Источник     : firefox-128.11.0-1.el10_0.inferit.src.rpm
Репозиторий  : @System
Из репозитор : appstream
Краткое опис : Mozilla Firefox Web browser
URL          : https://www.mozilla.org/firefox/
Лицензия     : MPLv1.1 or GPLv2+ or LGPLv2+
Описание     : Mozilla Firefox is an open-source web browser, designed for standards
            : compliance, performance and portability.
```

Рассмотрим основные операции с модулями.

Посмотреть список доступных модулей

Для просмотра списка доступных модулей выполните следующую команду:

```
$ dnf module list
```

Установить выбранный модуль

Для установки выбранного модуля выполните следующую команду:

```
$ sudo dnf module install имя_модуля:версия
```

Например, для установки модуля `ruby:3.1` используйте следующую команду:

```
$ sudo dnf module install ruby:3.1
```

Удалить указанный модуль

Для удаления указанного модуля выполните следующую команду:

```
$ sudo dnf module remove имя_модуля:версия
```

Например, для удаления пакета `ruby:3.1` используйте следующую команду:

```
$ sudo dnf module remove ruby:3.1
```

Описание репозиториев ОС МСВСфера

Рассмотрим репозитории ОС МСВСфера.

- **MSVSphere - AppStream** — приложения общего назначения.
- **MSVSphere - BaseOS** — базовый набор пакетов операционной системы.
- **MSVSphere - CRB** — дополнительные пакеты для разработчиков.

- **MSVSphere - Extras** — набор дополнительных приложений.
- **MSVSphere - HighAvailability** — пакеты для создания кластеров высокой доступности.
- **MSVSphere - NFV** — компоненты для виртуализации сетевых служб.
- **MSVSphere - ResilientStorage** — пакеты для создания кластерных хранилищ.
- **MSVSphere - RT** — набор пакетов для системы реального времени.

Также обычно в названии репозитория указывается версия операционной системы.

Посмотреть список включённых и доступных репозиториев

Для просмотра списка включённых репозиториев выполните следующую команду:

```
$ dnf repolist
```

Для просмотра списка включённых и отключённых репозиториев выполните следующую команду:

```
$ dnf repolist all
```

Для вывода подробного описания для каждого включённого репозитория выполните следующую команду:

```
$ dnf repolist -v
```

Для вывода списка отключённых репозиториев выполните следующую команду:

```
$ dnf repolist disabled
```

Для получения подробной информации о конкретном репозитории выполните следующую команду:

```
$ dnf repolist название репозитория -v
```

Пример: вывести подробную информацию о репозитории BaseOS (для ОС МСВСфера 10):

```
$ dnf repolist BaseOS -v
...
...
ИД репозитория              : baseos
Имя репозитория             : MSVSphere 10 - BaseOS
Статус репозитория          : включено
Версия репозитория          : 10.0
Метки дистрибутива          : [cpe:/o:ncsd:msvsphere:10]:  , 10, M, S, S, V, e, e, h, p, r
Репозиторий обновлен        : Вт 01 июл 2025 15:33:16
Пакеты репозитория          : 1 214
Пакеты-в-репозитории        : 1 214
Размер-репозитория          : 4.8 G
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
Зеркала-репозитория        : https://mirrors.inferitos.ru/mirrorlist/10/baseos
Базовый-URL-репозитория    : https://repo1.msvsphere-os.ru/msvsphere/10/isos/x86_64/ (0 more)
Истечение срока репозитория: 86 400 секунд(а) (осталось: Ср 02 июл 2025 16:05:43)
Имя файла репозитория      : /etc/yum.repos.d/msvsphere-baseos.repo
Всего пакетов              : 1 214
```

Здесь мы видим, что репозиторий включён, количество пакетов в репозитории и его размер, а также другие важные параметры.

Зеркала репозитория — это серверы, дублирующие содержимое этого репозитория. Они позволяют снизить нагрузку с основных серверов.

Добавить в систему сторонний репозиторий

Иногда возникает необходимость установить приложение, которого нет в имеющихся репозиториях. В этом случае есть возможность добавить в систему сторонний репозиторий.

Важно

Рекомендуем быть предельно осторожными при подключении сторонних репозиториев и тщательно соблюдать меры безопасности.

Вы можете подключить сторонний репозиторий, если есть `.repo`-файл, с помощью следующей команды:

```
$ dnf config-manager --add-repo путь_к_.repo_файлу
```

Пример подключения `.repo`-файла docker.io:

```
$ dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

Включить или отключить репозиторий

Вы можете по необходимости временно включать и отключать репозитории, чтобы установить приложение из конкретного репозитория. При этом репозиторий не будет удалён.

Команда включения репозитория:

```
$ sudo dnf config-manager --set-enabled имя_репозитория
```

Команда отключения репозитория:

```
$ sudo dnf config-manager --set-disabled имя_репозитория
```

При необходимости вы можете вывести справку по команде `config-manager`:

```
$ dnf config-manager --help-cmd
```

Безопасность

Использование сторонних репозиториев/пакетов

Так как сторонние репозитории и пакеты загружаются из Интернета, то при их скачивании и установке необходимо быть уверенными в безопасности устанавливаемых приложений. Важно быть уверенным, что никакая третья сторона не изменяла содержимое пакета при передаче его от автора к пользователю. Подписание пакета является способом защиты пакета для конечного пользователя. Поэтому репозитории и все пакеты в них подписываются специальным цифровым ключом.

Приватный ключ есть только у разработчиков. Публичный ключ может располагаться на сайте репозитория, либо распространяться вместе с операционной системой.

Разработчики подписывают пакеты приватным ключом, а с помощью публичного ключа конечный пользователь может убедиться, что это тот самый пакет и никакая третья сторона не изменяла его.

Ниже мы рассмотрим, как проверить цифровую подпись пакета.

Цифровые подписи пакетов и их проверка

Для проверки цифровой подписи пакета выполните следующую команду (находясь в папке с пакетом):

```
$ rpm --checksig имя_пакета.rpm
```

Пример: проверим цифровую подпись пакета `VirtualBox-7.1-7.1.10_169112_el10-1.x86_64.rpm`:

```
$ rpm --checksig VirtualBox-7.1-7.1.10_169112_el10-1.x86_64.rpm
VirtualBox-7.1-7.1.10_169112_el10-1.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

Вы можете также использовать опцию `-v` для вывода более полной информации о проверке.

Графический интерфейс к менеджеру пакетов DNF dnfdragora

Введение

dnfdragora — это удобный графический интерфейс к менеджеру пакетов DNF. Он объединяет надёжность командных инструментов с наглядностью современного

приложения, позволяя управлять программами и обновлениями в несколько кликов — без необходимости запоминать команды и параметры.

Этот инструмент подходит как для новичка, так и для опытного администратора: первый получает простую и безопасную навигацию по пакетам, второй — быстрые операции и полный контроль над тем, что и как будет установлено.

С помощью dnfdragora вы можете:

- находить и изучать пакеты по названию, описанию и группам;
- устанавливать, обновлять и удалять приложения и компоненты системы;
- видеть заранее, какие зависимости будут затронуты и какой объём данных потребуется;
- гибко фильтровать результаты (по статусу, репозиториям и т.д.);
- управлять источниками пакетов (репозиториями) и просматривать историю действий.

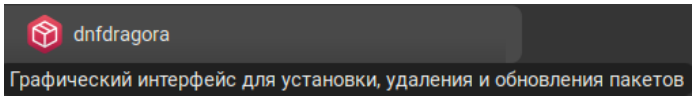
Установка

Для установки dnfdragora выполните в «Терминале» следующую команду:

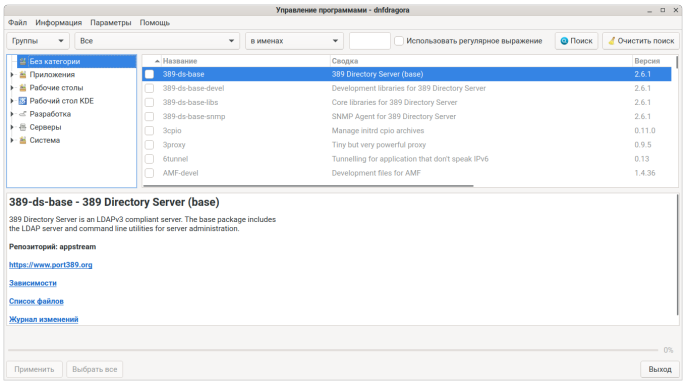
```
$ sudo dnf install -y dnfdragora
```

Использование

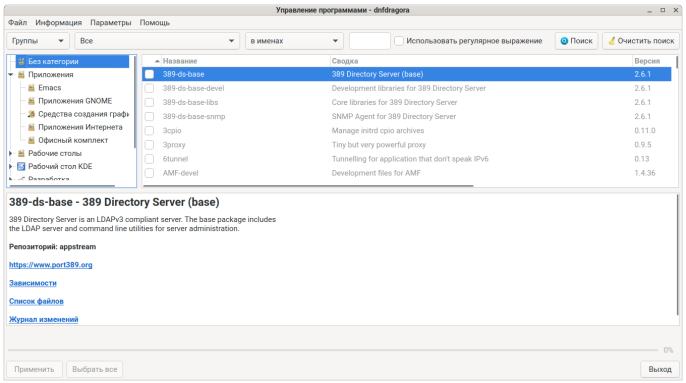
После установки dnfdragora появится в Arc menu:



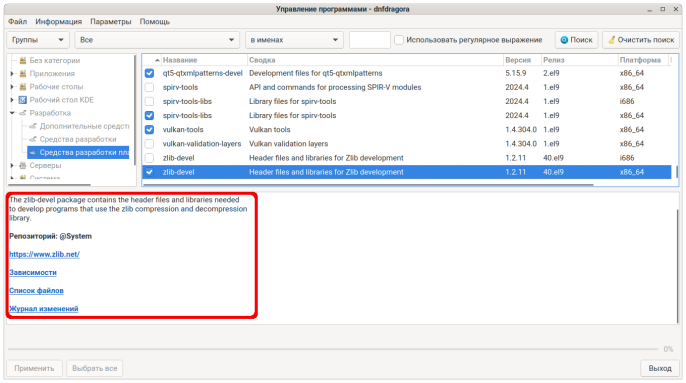
При запуске приложение будет анализировать репозитории, которые подключены к ОС, что может занять какое-то время. Как только списки всех пакетов в репозиториях будут сформированы, dnfdragora будет иметь следующий вид:



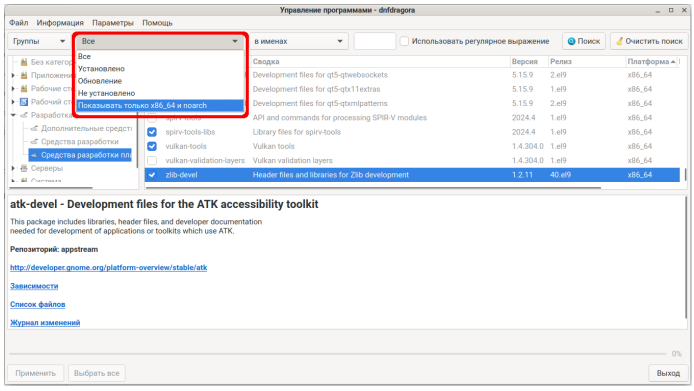
dnfdragora автоматически группирует приложения в «Категории». Это упрощает навигацию по пакетам.



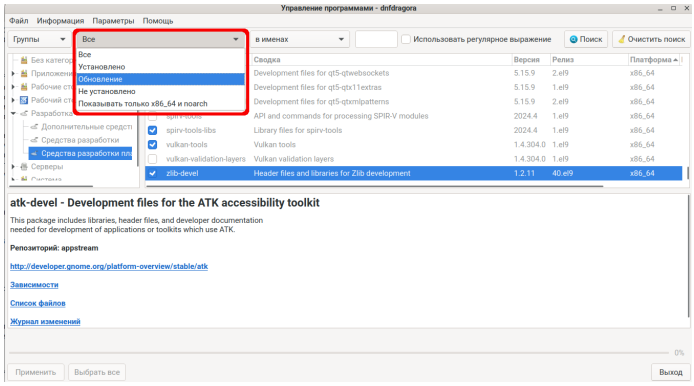
Выбрав нужный пакет можно посмотреть его зависимости, список файлов внутри исполняемого файла, журнал изменений, описание пакета и даже ссылку на сайт разработчика.



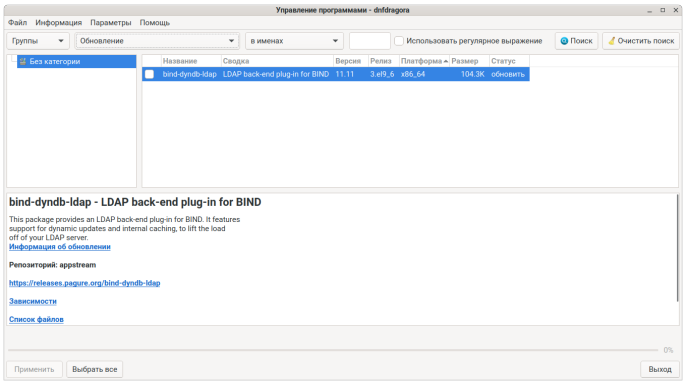
По стандарту dnfdragora выводит архитектуры: `i686`, `x86_64`, `noarch`. Чтобы отображать в списках только `x86_64` и `noarch` нажмите на «Все», а после выберите «Показывать только x86_64 и noarch».



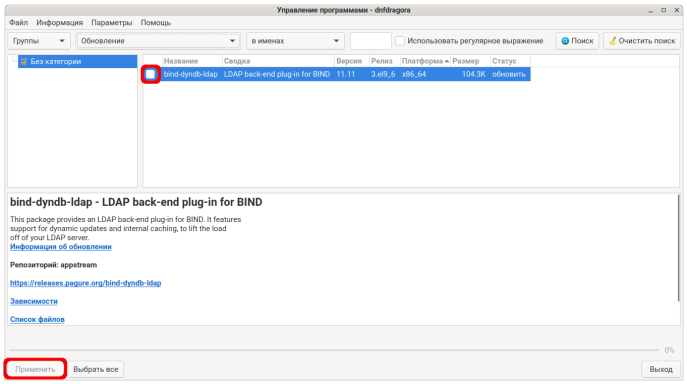
Чтобы dnfdragora показывал только обновления, в графическом интерфейсе переключитесь только на обновления. Для этого выберите пункт «Обновление».



Выбрав «Обновление» на экране будут только пакеты, которые можно обновить:

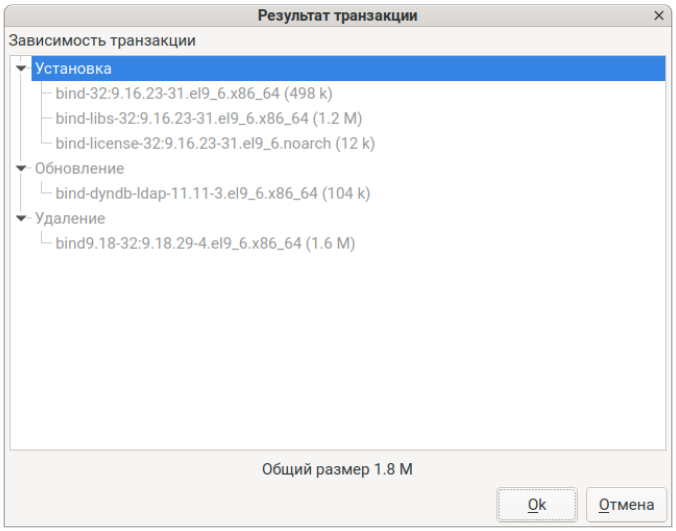


Чтобы установить выбранный пакет из списка, отметьте его. Теперь пакет считается помеченным на то или иное действие — в данном случае на обновление. Далее нажмите кнопку «Применить».

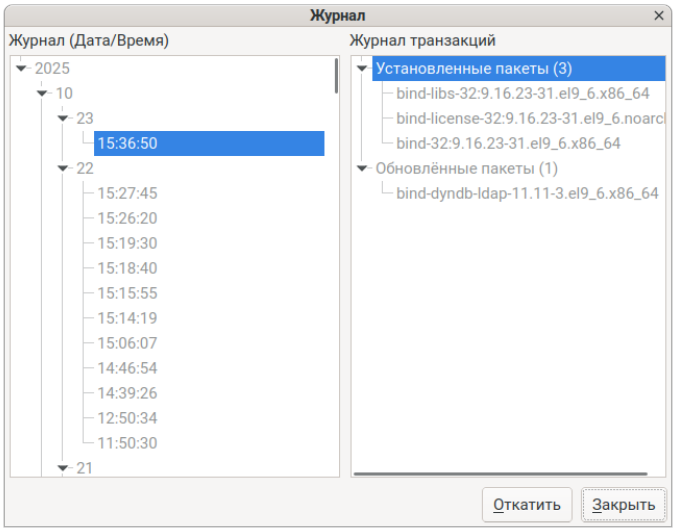


dnfdragora покажет полный список изменений в пакетах. Ознакомьтесь и нажмите «Ok». Подтвердив обновление пакета, приложение запросит пароль учётной записи

пользователя, а после завершения обновления выведет информацию об успешном или не успешном обновлении пакета.



Также приложение предоставляет возможность в удобном формате просматривать историю обновления/установки/удаления пакетов. Чтобы посмотреть историю перейдите в «Информация» → «История»:



Идентификация и аутентификация

Введение

Средства идентификации и аутентификации предоставляют возможности идентификации объектов доступа, идентификации и проверки подлинности субъектов доступа при входе в систему и при доступе к защищаемым объектам, управления идентификаторами, в том числе их создания, присвоения и уничтожения, управления аутентификационными данными, в том числе их инициализации, защищенного хранения, блокирования и разблокирования, проверки соответствия аутентификационной информации заданной метрике качества, защиты обратной связи при вводе аутентификационной информации, а также другие возможности.

Добавление нового пользователя

Для добавления нового пользователя используется утилита `useradd`. Она позволяет добавить учетную запись нового пользователя. Режимы ее работы и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 3: Опции утилиты `useradd` и их значения

Опция	Значение
`-c, --comment`	Любая текстовая строка. Используется как поле для имени и фамилии пользователя, длина этого поля не должна превосходить 128 символов.
`-b, --base-dir`	Базовый системный каталог по умолчанию, если не указан другой каталог. Базовый каталог объединяется с именем учётной записи для определения домашнего каталога.
`-d, --home`	Для создаваемого пользователя в качестве начального каталога будет использован базовый каталог. По умолчанию это значение получается объединением имени пользователя с базовым каталогом и используется как имя домашнего каталога.
`-d, --home-dir`	Задать домашний каталог нового пользователя. Если данная опция не используется, то в качестве домашнего каталога выбирается каталог типа `/базовый_системный_каталог/имя_пользователя`.
`-D, --defaults`	Вывести значения стандартных опций.
`-e, --expiredate`	Дата окончания срока действия учётной записи пользователя. Задаётся в формате `ГГГГ-ММ-ДД`.
`--f, --inactive`	Число дней, которые должны пройти после окончания срока действия пароля, чтобы учётная запись заблокировалась. Если указано значение `0`, то учётная запись блокируется сразу после окончания срока действия пароля, а при значении `-1` данная возможность не используется. По умолчанию используется значение `-1`.
`-g, --gid`	Название группы нового пользователя или её идентификационный номер. Указываемое название группы или её номер должны существовать в системе.
`-G, --groups`	Список дополнительных групп, в которых числится пользователь. Перечисление групп осуществляется через запятую без пробелов. На указанные группы действуют те же ограничения, что и для группы, указанной в опции `-g`.
`-m, --create-home`	Создает начальный домашний каталог нового пользователя, если он ещё не существует. Если каталог уже существует, добавляемый пользователь должен иметь права на доступ к указанному каталогу.
`-M, --no-create-home`	Позволяет не создавать домашний каталог нового пользователя.
`-K, --key`	Используется для изменения значений по умолчанию для параметров, хранимых в конфигурационном файле `/etc/login.def`.
`-N, --no-user-group`	Позволяет добавить нового пользователя в группу, указанную в опции `-g` или заданную по умолчанию в конфигурационном файле `/etc/default/useradd`, не создавая группу, название которой совпадает с именем нового пользователя. Если опции `-g`, `-N`, `-U` не указаны, то настройки групп по умолчанию определяются в конфигурационном файле `/etc/login.defs`.
`-o, --non-unique`	Позволяет создать учётную запись с уже имеющимся, не уникальным идентификатором.
`-p, --password`	Позволяет задать новый пароль для учётной записи.
`-r, --system`	Позволяет создать системную учётную запись. По умолчанию для данной категории учетных записей домашний каталог не создаётся вне зависимости от значения соответствующего параметра конфигурационного файла `/etc/login.defs`. Для создания домашнего каталога системного пользователя необходимо вместе с опцией `-r` задать опцию `-m`.
`-s, --shell`	Полный путь к программе, используемой в качестве начального командного интерпретатора для пользователя сразу после регистрации. Длина этого поля не должна превосходить 256 символов. Если задать пустое значение, то будет использоваться оболочка по умолчанию.
`-u, --uid`	Позволяет задать идентификационный номер (численное неотрицательное значение идентификатора) пользователя. Это значение должно быть уникальным, если не задействована опция `-o`.
`U, --user-group`	Позволяет создать группу, название которой совпадает с именем пользователя, присоединив данного пользователя к этой группе.
`-h, --help`	Показать краткую справку об утилите.

Пример: создадим пользователя с именем `user` и зададим для него основную группу `users` и две дополнительные группы `ftp` и `developers`, к которым он будет приписан.

Для этого выполним следующую команду:

```
$ sudo useradd -g users -G ftp,developers user
```

Изменение уже имеющихся пользовательских записей

Для изменения уже имеющихся пользовательских записей используется утилита `usermod`. Она позволяет изменить данные существующей учётной записи пользователя. Режимы ее работы и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 4: Опции утилиты `usermod` и их значения

Опция	Значение
`-a, --append`	Добавить пользователя в дополнительную группу. Следует использовать только вместе с параметром `-G`.
`-c, --comment`	Новое значение поля комментария.
`d, --home`	Новый домашний каталог учётной записи. Если указан параметр `-m`, то содержимое текущего домашнего каталога будет перемещено в новый домашний каталог, который будет создан, если он ещё не существует.
`-e, --expiredate`	Установить дату окончания срока действия учётной записи в формате `ГГГГ-ММ-ДД`.
`-f, --inactive`	Установить пароль после окончания срока действия учётной записи в `INACTIVE`. Если указано значение `0`, то учётная запись блокируется сразу после окончания срока действия пароля, а при значении `-1` данная возможность не используется. По умолчанию используется значение `-1`.
`-g, --gid`	Принудительно назначить первичную группу.
`-G, --groups`	Список дополнительных групп.
`-l, --login`	Новое значение учётной записи.
`-L, --lock`	Заблокировать пароль пользователя. Это делается помещением символа `!` в начало шифрованного пароля, что приводит к его блокировке. Не следует использовать этот параметр вместе с `-p` или `-U`.
`-m, --move-home`	Переместить содержимое домашнего каталога пользователя в новое место. Если новый домашний каталог не существует, то он создаётся автоматически. Данная опция используется только вместе с опцией `-d`.
`-o, --non-unique.`	При использовании с параметром `-u` этот параметр позволяет указывать не уникальный числовой идентификатор пользователя.
`-p, --password`	Задать новый пароль для учётной записи.
`-s, --shell`	Задать новую оболочку для учётной записи.
`-u, --uid`	Новый идентификационный номер для учётной записи.
`-U, --unlock`	Разблокировать учетную запись.

Пример: изменим срок действия учётной записи пользователя с идентификатором `user6`.

Для этого выполним следующую команду:

```
$ sudo usermod -e 2020-05-01 user6
```

где 2020-05-01 — дата истечения срока действия учётной записи в формате `ГГГГ-ММ-ДД`.

Пример: изменим идентификатор (значение учётной записи) пользователя с `user6` на `user7`.

Для этого выполним следующую команду:

```
$ sudo usermod -l user7 user6
```

Удаление пользователей

Для удаления пользователей используется утилита `userdel`. Она позволяет удалить существующую учетную запись пользователя. Режимы ее работы и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 5: Опции утилиты `userdel` и их значения

Опция	Значение
`-f, --force`	С этой опцией учётная запись будет удалена, даже если пользователь в этот момент работает в системе. Она также заставляет утилиту удалить домашний каталог пользователя и почтовый ящик, даже если другой пользователь использует тот же домашний каталог или если почтовый ящик не принадлежит данному пользователю. **Внимание! Перед использованием этого параметра убедитесь в необходимости этого действия! Этот параметр может привести систему в нерабочее состояние!**
`-r, --remove`	Файлы в домашнем каталоге пользователя будут удалены вместе с самим домашним каталогом и почтовым ящиком. Пользовательские файлы, расположенные в других файловых системах, нужно искать и удалять вручную.
`-n`	Задает, сколько месяцев идентификатор пользователя должен устаревать перед повторным использованием. Задайте `-1`, чтобы указать, что идентификатор пользователя никогда не должен использоваться повторно. Задайте `0`, чтобы указать, что идентификатор пользователя можно немедленно использовать повторно. Если опция `-n` не задана, то идентификатор будет устаревать стандартное количество месяцев перед повторным использованием.
`-h, --help`	Показать краткую справку.

Пример: удалим пользователя с идентификатором `user7`.

Для этого выполним следующую команду:

```
$ sudo userdel -r user7
```

Добавление группы пользователей

Для добавления группы пользователей используется утилита `groupadd`. Режимы ее работы и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 6: Опции утилиты `groupadd` и их значения

Опция	Значение
`-f`	Вернуть статус успешного выполнения, если группа уже существует. Если используется вместе с параметром `-g` и указанный идентификатор группы уже существует, то выбирается другой уникальный идентификатор группы, то есть параметр `-g` игнорируется.
`-g`	Числовое значение идентификатора группы. Значение должно быть уникальным, если не задан параметр `-o`. Значение должно быть не отрицательным. По умолчанию берётся значение больше 999 и больше идентификатора любой другой группы. Значения от 0 и до 999 обычно зарезервированы под системные группы.
`-K`	Изменить значения по умолчанию для параметров, которые хранятся в конфигурационном файле `/etc/login.defsм`.
`-o`	Разрешить добавление группы с не уникальным идентификатором.
`-r, --system`	Создать системную группу.
`-h, --help`	Показать краткую справку.

Пример: создадим группу `group2` с числовым значением идентификатора `8285`.

Для этого выполним следующую команду:

```
$ sudo groupadd group2 -g 8285
```

Изменение существующей группы пользователей

Для изменения существующей группы пользователей используется утилита `groupmod`. Режимы ее работы и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 7: Опции утилиты `groupmod` и их значения

Опция	Значение
`-g, --gid`	Изменить идентификатор группы.
`-n, --new-name`	Изменить имя группы.
`-o, --non-unique`	Позволяет использовать не уникальный идентификатор группы.
`-p, --password`	Изменить пароль.
`-h, --help`	Показать краткую справку.

Пример: изменим идентификатор группы пользователей `users` на `ftp`.

Для этого выполним следующую команду:

```
$ sudo groupmod -g ftp users
```

Удаление существующей группы пользователей

Для удаления существующей группы пользователей используется утилита `groupdel`. Утилита позволяет удалить определение группы из системы путем удаления записи о соответствующей группе из файла `/etc/group`. Однако она не удаляет идентификатор группы из файла паролей. Удаленный идентификатор действует для всех файлов и каталогов, которые его имели.

Пример: удалим группу с именем `group3`.

Для этого выполним следующую команду:

```
$ sudo groupdel group3
```

Создание и изменение пароля пользователя

Для создания и изменения пароля пользователя (в том числе для блокировки учётной записи пользователя) используется утилита `passwd`. Обычный пользователь может изменить пароль только своей учётной записи, суперпользователь `root` может изменить пароль любой учётной записи.

При изменении пароля проверяется информация об устаревании пароля, чтобы убедиться, что пользователю разрешено изменять пароль в настоящий момент. Если выяснится, что не разрешено, то утилита не производит изменение пароля и завершает работу.

При изменении пароля пользователь должен будет сначала ввести старый пароль, если он был. Введенное пользователем значение старого пароля зашифровывается и сравнивается со значением зашифрованного текущего пароля. Затем пользователю необходимо будет дважды ввести новый пароль. Значение второго ввода сравнивается с первым, и они должны совпасть. После этого пароль тестируется на сложность подбора, т.е. его значение не должно быть легко угадываемым.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 8: Опции утилиты `passwd` и их значения

Опция	Значение
`-a, --all`	Эту опцию можно использовать только вместе с `-S` для вывода статуса всех пользователей.
`-d, --delete`	Удалить пароль пользователя (сделать его пустым). Это быстрый способ заблокировать пароль учётной записи.
`-e, --expire`	Немедленно сделать пароль устаревшим. Это заставит пользователя изменить пароль при следующем входе в систему.
`-i, --inactive`	Эта опция используется для блокировки учётной записи по прошествии заданного числа дней после устаревания пароля. То есть если пароль устарел и прошло больше дней, чем указано, то пользователь больше не сможет использовать свою учётную запись.
`-l, --lock`	Заблокировать указанную учётную запись. Эта опция блокирует учётную запись путем изменения значения пароля на такое, которое не может быть ранее указанным зашифрованным паролем.
`-m, --mindays`	Задать минимальное количество дней между сменой пароля. Нулевое значение этого поля указывает на то, что пользователь может менять свой пароль тогда, когда захочет.
`-S, --status`	Показать состояние учётной записи. Информация о состоянии содержит семь полей. Первое поле содержит имя учётной записи. Второе поле указывает, заблокирована ли учётная запись, она без пароля или у неё есть рабочий пароль. Третье поле хранит дату последнего изменения пароля. В следующих четырёх полях хранятся минимальный срок, максимальный срок, период выдачи предупреждения и период неактивности пароля. Все эти сроки измеряются в днях.
`-u, --unlock`	Разблокировать указанную учётную запись. Этот параметр активирует учётную запись путем изменения пароля на прежнее значение, которое было перед использованием параметра `-l`.
`-w, --warndays`	Установить число дней выдачи предупреждения, перед тем как потребуется смена пароля.
`-x, --maxdays`	Установить максимальное количество дней, в течение которых пароль остаётся рабочим, после чего его надо будет изменить.
`-h, --help`	Показать краткую справку.

Пример: зададим пароль пользователю `user4`. Работа команды `passwd`:

```
$ sudo passwd user4
Изменяется пароль пользователя user4.
Новый пароль :
Повторите ввод нового пароля :
passwd: все данные аутентификации успешно обновлены.
```

Пример: посмотрим состояние учётной записи `user4`. Работа команды `passwd`:

```
$ sudo passwd -S user4
user4 PS 2023-07-04 0 99999 7 -1 (Пароль задан, шифр SHA512.)
```

Где:

- `user4` — имя пользователя.
- `PS` — статус пароля.
- `2023-07-04` — отображает время последнего изменения пароля.
- `0` и `99999` — минимальный и максимальный срок действия пароля.
- `7` — срок вывода предупреждения.

- `-l` — срок деактивации пароля.

Изменение срока действия учётной записи и пароля пользователя

Утилита `chage` позволяет установить дату завершения срока действия учётной записи пользователя, минимальный и максимальный срок действия пароля, дату завершения срока действия пароля, а также количество дней, в течение которых пользователю будут выводиться предупреждения о приближении завершения срока действия пароля.

Командой `chage` может пользоваться только суперпользователь, за исключением использования её с параметром `-l`, который позволяет непривилегированным пользователям определить время, когда истекает их личный пароль или учетная запись.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 9: Опции утилиты `chage` и их значения

Опция	Значение
`-m`	Меняет значение `mindays` на минимальное число дней между сменой пароля. Значение `0` в этом поле обозначает, что пользователь может изменять свой пароль когда угодно.
`-M`	Меняет значение `maxdays` на максимальное число дней, в течение которых пароль будет действителен. Когда сумма `maxdays` и `lastday` меньше, чем текущий день, у пользователя будет запрошен новый пароль до начала работы в системе.
`-d`	Меняет значение `lastday` на день, когда пароль был изменен последний раз (число дней с 1 января 1970). Дата также может быть указана в формате `ГГГГ-ММ-ДД`.
`-E`	Используется для задания даты, с которой учетная запись пользователя станет недоступной. Дата также может быть указана в формате `ГГГГ-ММ-ДД`.
`-I`	Используется для задания количества дней «неактивности», то есть дней, когда пользователь вообще не входил в систему, после которых его учетная запись будет заблокирована. Значение `0` отключает этот режим.
`-W`	Используется для задания числа дней, когда пользователю начнет выводиться предупреждение об истечении срока действия его пароля и необходимости его изменения.
`-l`	Просмотреть текущую информацию о дате истечения срока действия пароля для пользователя.

Пример: просмотрим текущую информацию о дате истечения срока действия пароля для пользователя `user4`. Работа команды `chage`:

```
$ sudo chage -l user4
Последний раз пароль был изменён: мар 12, 2023
Срок действия пароля истекает: никогда
Пароль будет деактивирован через: никогда
Срок действия учётной записи истекает: никогда
Минимальное количество дней между сменой пароля: 0
Максимальное количество дней между сменой пароля: 99999
Количество дней с предупреждением перед деактивацией пароля: 7
```

Управление политиками паролей

В данном разделе описана процедура управления политиками паролей на локальной системе, не подключённой к LDAP-каталогу пользователей (FreeIPA, Microsoft

Active Directory и т.д.). В случае использования LDAP-каталога обратитесь к соответствующему руководству по администрированию.

Для управления политиками паролей в ОС МСВСфера используется системная утилита `authselect`, которая оперирует профилями аутентификации. В первую очередь необходимо убедиться, что выбран профиль аутентификации. Вы можете это сделать с помощью команды `authselect current`:

```
# вывод для систем, подключённых к каталогу пользователей FreeIPA
$ sudo authselect current
Profile ID: sssd
Enabled features:
- with-mkhomedir
- with-sudo

# вывод для систем, использующих профиль "minimal"
$ sudo authselect current
Profile ID: minimal
Enabled features: None

# вывод для систем, не использующих профиль authselect
$ sudo authselect current
Конфигурация не обнаружена. / No existing configuration detected.
```

Если система не настроена на использование профиля аутентификации, то необходимо выбрать его, чтобы получить возможность настраивать политики паролей.

Просмотреть список доступных профилей `authselect` вы можете следующим образом:

```
$ sudo authselect list
- minimal              Local users only for minimal installations
- sssd                 Enable SSSD for system authentication (also for local users only)
- winbind              Enable winbind for system authentication
- custom/minimal_gost  Local users only for minimal installations and gost support
- custom/sssd_gost     Enable SSSD with GOST support for system authentication (also for local␣
↪users only)
```

Выбрать профиль вы можете с помощью команды `authselect select`. Для локальной системы рекомендуется использовать профиль `minimal`:

```
$ sudo authselect select minimal --force
[error] File [/etc/authselect/system-auth] is still present
[error] File [/etc/authselect/password-auth] is still present
[error] File [/etc/authselect/fingerprint-auth] is still present
[error] File [/etc/authselect/smartcard-auth] is still present
[error] File [/etc/authselect/postlogin] is still present
[error] File [/etc/authselect/nsswitch.conf] is still present
[error] File [/etc/authselect/dconf-db] is still present
[error] File [/etc/authselect/dconf-locks] is still present
[error] Link [/etc/pam.d/system-auth] points to [/etc/authselect/system-auth]
[error] Symbolic link [/etc/pam.d/system-auth] to [/etc/authselect/system-auth] still exists!
[error] Link [/etc/pam.d/password-auth] points to [/etc/authselect/password-auth]
[error] Symbolic link [/etc/pam.d/password-auth] to [/etc/authselect/password-auth] still exists!
[error] Link [/etc/pam.d/fingerprint-auth] points to [/etc/authselect/fingerprint-auth]
[error] Symbolic link [/etc/pam.d/fingerprint-auth] to [/etc/authselect/fingerprint-auth] still␣
↪exists!
[error] Link [/etc/pam.d/smartcard-auth] points to [/etc/authselect/smartcard-auth]
[error] Symbolic link [/etc/pam.d/smartcard-auth] to [/etc/authselect/smartcard-auth] still␣
↪exists!
[error] Link [/etc/pam.d/postlogin] points to [/etc/authselect/postlogin]
[error] Symbolic link [/etc/pam.d/postlogin] to [/etc/authselect/postlogin] still exists!
[error] Link [/etc/nsswitch.conf] points to [/etc/authselect/nsswitch.conf]
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
[error] Symbolic link [/etc/nsswitch.conf] to [/etc/authselect/nsswitch.conf] still exists!
[error] Link [/etc/dconf/db/distro.d/20-authselect] points to [/etc/authselect/dconf-db]
[error] Symbolic link [/etc/dconf/db/distro.d/20-authselect] to [/etc/authselect/dconf-db] still␣
↪exists!
[error] Link [/etc/dconf/db/distro.d/locks/20-authselect] points to [/etc/authselect/dconf-locks]
[error] Symbolic link [/etc/dconf/db/distro.d/locks/20-authselect] to [/etc/authselect/dconf-
↪locks] still exists!
Backup stored at /var/lib/authselect/backups/2024-10-08-16-28-25.N8QZyv
Profile "minimal" was selected.
The following nsswitch maps are overwritten by the profile:
- aliases
- automount
- ethers
- group
- hosts
- initgroups
- netgroup
- networks
- passwd
- protocols
- publickey
- rpc
- services
- shadow
```

Проверить корректность применения профиля вы можете следующим образом:

```
$ sudo authselect current
Profile ID: minimal
Enabled features: None

$ sudo authselect check
Current configuration is valid.
```

Управление требованиями к качеству паролей

По умолчанию ОС МСВСфера предъявляет следующие требования к качеству паролей пользователя:

- пароль должен иметь длину как минимум 8 символов;
- пароль должен отсутствовать в словаре известных паролей программы `cracklib`.

За проверку качества паролей отвечает PAM-модуль `pam_pwquality`, который включён по умолчанию для всех профилей аутентификации.

Модуль настраивается через конфигурационный файл `/etc/security/pwquality.conf`, который по умолчанию имеет следующий вид (для переменных указаны значения по умолчанию, описание параметров переведено на русский язык и добавлены комментарии):

```
# Количество символов в новом пароле, которые не должны присутствовать в старом
# пароле. Значение 0 полностью отключает проверку на пересечение символов, за
# исключением попытки использования идентичного пароля.
# difok = 1

# Минимально допустимое количество символов в новом пароле (плюс один, если
# использование кредитов не отключено, что является поведением по умолчанию).
# Пароль не может быть короче 6 символов.
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# minlen = 8

# Максимальное количество кредитов, начисляемое за наличие цифр в новом пароле.
# Если значение меньше 0, то это минимальное количество цифр в новом пароле.
# dcredit = 0

# Максимальное количество кредитов, начисляемое за наличие прописных букв в
# новом пароле. Если значение меньше 0, то это минимальное количество прописных
# букв в новом пароле.
# ucredit = 0

# Максимальное количество кредитов, начисляемое за наличие строчных букв в
# новом пароле. Если значение меньше 0, то это минимальное количество строчных
# букв в новом пароле.
# lcredit = 0

# Максимальное количество кредитов, начисляемое за наличие других символов в
# новом пароле. Если значение меньше нуля, то это минимальное количество других
# символов в новом пароле.
# ocredit = 0

# Минимальное количество требуемых классов символов в новом пароле
# (цифры, буквы в нижнем регистре, буквы в верхнем регистре, другие символы).
# minclass = 0

# Максимальное количество разрешённых повторяющихся символов в новом пароле.
# Проверка отключается, если значение равно 0.
# maxrepeat = 0

# Максимальное количество повторяющихся символов из одного класса, разрешённое
# в новом пароле. Проверка отключается, если значение равно 0.
# maxclassrepeat = 0

# Проверять, есть ли слова из поля GECOS пользователя в новом пароле. Проверка
# отключается, если значение равно 0.
# gecoscheck = 0

# Проверять наличие пароля в словаре cracklib, если значение не равно 0.
# dictcheck = 1

# Проверять наличие имени пользователя в новом пароле. Проверка отключается,
# если значение равно 0.
# usercheck = 1

# Длина подстрок из имени пользователя, которую нужно проверить на наличие
# в новом пароле. Эта проверка выполняется, если значение больше 0 и значение
# usercheck равно 1.
# usersubstr = 0

# Включить принудительную проверку нового пароля пользователя, новый пароль,
# не соответствующий требованиям, будет отклонён, если значение не равно 0.
# enforcing = 1

# Путь к словарям cracklib. Если не задан, будет использован стандартный
# словарь cracklib.
# dictpath =

# Сколько раз запрашивать новый пароль пользователя прежде чем выводить ошибку.
# Значение по умолчанию - 1.
# retry = 3

# Применять требования к качеству пароля пользователя root, если опция
# раскомментирована.
# enforce_for_root

# Применять требования к качеству пароля только для локальных пользователей,
# присутсвующих в файле /etc/passwd, если эта опция раскомментирована.
# local_users_only
```

Кредиты в параметрах `dcredit`, `ucredit`, `lcredit` и `ocredit` определяют сколько баллов может быть начислено за определённый тип символов, используемый в новом пароле. Если значение параметра больше `0`, то за каждый такой символ к общей длине пароля добавляется определённое количество кредитов.

Пример: если все четыре параметра имеют значение `1` и минимально допустимая длина пароля составляет 7 символов, то при использовании всех 4 типов символов потребуется ввести пароль всего лишь из 7 символов. За каждый неиспользованный тип символа к требуемой длине пароля добавляется штраф, указанный в соответствующем параметре. Так, если не использовать цифры и строчные буквы, то минимальная длина пароля уже составит 9 символов.

Использование механизма кредитов позволяет ослабить требования к длине пароля за счёт использования символов из разных групп.

После внесения правок в конфигурационный файл `/etc/security/pwquality.conf` перезапуск каких-либо сервисов не требуется — PAM применит изменения автоматически и новые настройки будут использованы при следующем изменении пароля.

Дополнительную информацию по использованию модуля `pam_pwquality` вы можете найти в соответствующих руководствах:

- `man pam_pwquality`;
- `man pwquality.conf`.

Ограничение на повторное использование паролей

В конфигурации по умолчанию операционная система не позволяет повторно использовать только текущий пароль пользователя. Это поведение можно изменить с помощью PAM-модуля `pam_pwhistory`, который позволяет хранить историю паролей для каждого пользователя.

Для включения модуля `pam_pwhistory` выполните следующую команду:

```
$ sudo authselect enable-feature with-pwhistory
```

После этого свойство `with-pwhistory` должно появиться в свойствах текущего профиля аутентификации:

```
$ sudo authselect current
Profile ID: minimal
Enabled features:
- with-faillock
- with-pwhistory
```

Модуль настраивается через конфигурационный файл `/etc/security/pwhistory.conf`, который по умолчанию имеет следующий вид (для переменных указаны значения по умолчанию, описание параметров переведено на русский язык и добавлены комментарии):

```
# Раскомментирование этой опции включает вывод отладочной информации.
# debug

# Так же сохранять предыдущие пароли пользователя root, если эта опция
# раскомментирована.
# enforce_for_root

# Количество сохраняемых паролей для каждого пользователя.
# remember = 10

# Сколько раз запрашивать новый пароль пользователя прежде чем выводить ошибку.
# retry = 1

# Каталог, в котором будут храниться предыдущие пароли пользователей.
# file = /etc/security/opasswd
```

Исходя из описания выше, после включения модуля `pam_pwhistory`, система будет хранить последние 10 паролей для каждого пользователя и выдавать ошибку при попытке использовать сохранённый пароль в качестве нового при смене пароля.

Пример ошибки:

```
# на английском языке
$ passwd
Changing password for user test.
Current password:
New password:
Retype new password:
Password has been already used. Choose another.
passwd: Have exhausted maximum number of retries for service

# на русском языке
$ passwd
Изменение пароля пользователя test.
Текущий пароль:
Новый пароль:
Повторите ввод нового пароля:
Этот пароль уже был использован. Выберите другой.
```

Дополнительную информацию по использованию модуля `pam_pwhistory` вы можете найти в соответствующих руководствах:

- `man pam_pwhistory`;
- `man pwhistory.conf`.

Ограничение количества неуспешных попыток аутентификации

В конфигурации по умолчанию операционная система не ограничивает количество неуспешных попыток аутентификации пользователя. Однако, реализация такого ограничения возможна с помощью PAM-модуля `pam_faillock`.

Для включения модуля `pam_faillock` выполните следующую команду:

```
$ sudo authselect enable-feature with-faillock
```

После этого свойство `with-faillock` должно появиться в свойствах текущего профиля аутентификации:

```
$ sudo authselect current
Profile ID: minimal
Enabled features:
- with-faillock
```

Модуль настраивается через конфигурационный файл `/etc/security/faillock.conf`, который по умолчанию имеет следующий вид (для переменных указаны значения по умолчанию, описание параметров переведено на русский язык и добавлены комментарии):

```
# Каталог, в котором хранятся файлы с записями об ошибках аутентификации
# пользователей.
# Внимание: в случае изменения этого пути потребуется дополнительно
# перенастроить правила SELinux.
# dir = /var/run/faillock

# Раскомментирование этой опции включает логирование имён несуществующих
# пользователей в системный журнал.
# audit

# Раскомментирование этой опции отключает вывод информационных сообщений на
# консоль.
# silent

# Раскомментирование этой опции отключает вывод информационных сообщений в
# системный журнал.
# no_log_info

# Раскомментирование этой опции включает отслеживание неудачных попыток
# аутентификации только для локальных пользователей, указанных в файле
# /etc/passwd. В таком случае пользователи из LDAP-каталога будут
# игнорироваться PAM-модулем "faillock". Включение данного параметра позволяет
# избежать ситуаций с двойной блокировкой, когда пользователь будет заблокирован
# и локально, и на уровне каталога пользователей.
# local_users_only

# Блокировать доступ пользователю, если количество последовальных ошибок
# аутентификации за последний промежуток времени превышает N попыток.
# deny = 3

# Временной интервал в секундах, в течении которого ошибки аутентификации
# пользователя считаются последовательными и приводят к блокировке его учётной
# записи. Значение по умолчанию — 15 минут.
# fail_interval = 900

# Автоматически восстановить доступ к заблокированной учётной записи спустя N
# секунд после блокировки. Для перманентной блокировки учётной записи задайте
# значение 0 — в таком случае восстановление доступа будет возможно только
# вручную через команду "faillock". Значение по умолчанию — 10 минут.
# unlock_time = 600

# Раскомментирование этой опции также приведёт к блокировке пользователя
# root в случае неудачных попыток аутентификации. По умолчанию блокируются
# только обычные пользователи.
# even_deny_root

# Автоматически восстанавливать доступ к заблокированной учётной записи
# пользователя root спустя N секунд после блокировки. Если значение не
# определено, модуль "faillock" будет использовать значение параметра "unlock_time".
# root_unlock_time = 900

# Имя группы системных администраторов. Если задано, то для участников этой
# группы будут применяться те же правила, что и для пользователя root
# (к ним будут применяться параметры "even_deny_root" и "root_unlock_time").
# admin_group = <admin_group_name>
```

Исходя из описания выше, при включении PAM-модуля `pam_faillock` будет

автоматически применена следующая конфигурация: локальные пользователи и пользователи из LDAP-каталога будут блокироваться после трёх неудачных попытках входа в течении 15 минут на срок в 10 минут. Пользователь `root` блокироваться не будет.

После внесения правок в конфигурационный файл `/etc/security/faillock.conf` перезапуск каких-либо сервисов не требуется — PAM применит изменения автоматически при обработке следующего запроса на аутентификацию.

Функциональность модуля `pam_faillock` распространяется как на локальную (через графическую или текстовую консоль), так и на сетевую (с помощью SSH) аутентификацию пользователей.

Для просмотра истории неудачных попыток аутентификации пользователя за учётный период `fail_interval` вы можете использовать следующую команду (замените «USERNAME» на имя реального пользователя из вашей системы):

```
$ sudo faillock --user USERNAME
When                Type  Source                                         Valid
2024-10-08 21:12:20 RHOST 192.168.1.4                                        V
2024-10-08 21:12:25 RHOST 192.168.1.4                                        V
2024-10-08 21:12:28 RHOST 192.168.1.4                                        V
```

Для снятия блокировки учётной записи пользователя вы можете использовать следуюзую команду:

```
$ sudo faillock --user USERNAME --reset
```

Дополнительную информацию по использованию модуля `pam_faillock` вы можете найти в соответствующих руководствах:

- `man pam_faillock`;
- `man faillock`;
- `man faillock.conf`.

Получение сведений о пользователе

Утилита `id` позволяет получить сведения об указанном пользователе или о текущем пользователе, запустившем данную утилиту, если он не указал явно имя пользователя.

По умолчанию выводятся числовые идентификаторы пользователя и группы, действующие идентификаторы пользователя и группы, а также идентификаторы других групп, в которых состоит пользователь.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 10: Опции утилиты `id` и их значения

Опция	Значение
`-g, --group`	Выводит только подлинный числовой идентификатор групп.
`-G, --groups`	Выводит все подлинные числовые идентификаторы групп, в которых состоит пользователь.
`-n, --name`	Выводит действующие имена пользователей или групп.
`-r, --real`	Выводит подлинные числовые идентификаторы пользователей или групп.
`-u, --user`	Выводит только подлинный числовой идентификатор пользователя.
`--version`	Выводит информацию о версии утилиты и завершает работу.
`--help`	Выводит справку по этой утилите и завершает работу.

Пример: выведем сведения о текущем пользователе `user`. Работа команды `id`:

```
$ id
uid=1000(user) gid=1000(user) группы=1000(user),100(users) контекст=user_u:user_r:user_t:s0
```

Конфигурационный файл /etc/login.defs

Конфигурационный файл `/etc/login.defs` позволяет задавать параметры, определяющие использование пользователями своих паролей.

```
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

Параметры перечислены в таблице:

Таблица 11: Параметры конфигурационного файла `/etc/login.defs` и их описание

Параметр	Описание
`PASS_MAX_DAYS`	Определяет максимальный срок действия пароля, т.е. максимальное число дней, в течение которых действие пароля сохраняется. По истечении этого срока запускается процесс принудительной смены пароля. Если значение параметра не задано, то есть параметр закомментирован символом `#` или ему присвоено значение `-1`, то данное ограничение не установлено (отменяется).
`PASS_MIN_DAYS`	определяет минимальный срок между изменениями пароля, т.е. минимальное число дней между двумя последовательными изменениями пароля. Если значение параметра не задано, то есть параметр закомментирован символом `#` или ему присвоено значение `-1`, то данное ограничение не установлено (отменяется).
`PASS_MIN_LEN`	Определяет минимальную допустимую длину задаваемого пароля.
`PASS_WARN_AGE`	Определяет, за сколько дней до истечения срока действия пароля начнётся вывод предупреждения о необходимости его смены. Если значение параметра не задано, то есть параметр закомментирован символом `#` или ему присвоено значение `-1`, то данное ограничение не установлено (отменяется). Если значение параметра `0`, то предупреждение о необходимости смены пароля будет выведено в день его устаревания.

Пример: выведем на экран текущее заданное значение максимального количества дней действия пароля:

```
$ cat /etc/login.defs | grep PASS_MAX_DAYS
# PASS_MAX_DAYS  Maximum number of days a password may be used.
PASS_MAX_DAYS 30
```

Мы видим, что текущее максимальное количество дней действия пароля — 30 дней.

Конфигурационный файл /etc/pam.d/system-auth

Конфигурационный файл `/etc/pam.d/system-auth` позволяет задавать настройки подключаемых модулей аутентификации.

```
# Generated by authselect on Fri Jul 14 14:08:55 2023
# Do not modify this file manually.

auth        required                                     pam_env.so
auth        required                                     pam_faildelay.so delay=20000
00
auth        sufficient                                   pam_fprintd.so
auth        [default=1 ignore=ignore success=ok]         pam_usertype.so isregular
auth        [default=1 ignore=ignore success=ok]         pam_localuser.so
auth        sufficient                                   pam_unix.so nullok
auth        [default=1 ignore=ignore success=ok]         pam_usertype.so isregular
auth        sufficient                                   pam_sss.so forward_pass
auth        required                                     pam_deny.so

account     required                                     pam_unix.so
account     sufficient                                   pam_localuser.so
account     sufficient                                   pam_usertype.so issystem
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required                                     pam_permit.so
```

Каждая строка в нем представляет собой правило, состоящее из трёх обязательных полей и одного опционального. Поля разделены символом пробела. Порядок, в котором указаны правила, определяет очередность их проверки.

Синтаксис правила:

```
type control module-path [module-arguments]
```

Поле `type` задаёт тип вызываемого модуля и может принимать одно из четырех допустимых значений:

- `auth` — предназначен для аутентификации пользователя путём запроса и проверки его пароля;
- `account` — используется для контроля доступа к сервису/приложению. Например, может быть произведён запрос о том, не истёк ли срок действия аккаунта пользователя, разрешено ли пользователю работать с определённым сервисом в определённое время, хватает ли системных ресурсов для работы;
- `password` — применятся для установки/изменения паролей;
- `session` — управляет действиями пользователя в рамках активной сессии после его успешной аутентификации в системе.

Поле `control` задаёт действие, которое нужно выполнить после вызова модуля. Доступно несколько действий:

- `required` — модуль должен вернуть положительный ответ. Если он возвращает отрицательный ответ, то пользователь будет уведомлен об этом только после того, как все остальные модули данного типа будут проверены;
- `requisite` — требует от модуля положительный ответ. В случае получения отрицательного ответа последовательная проверка выполнения остальных правил моментально прекращается и пользователь получает сообщение об ошибке аутентификации;
- `sufficient` — в случае, если ни один из модулей с действием `required` или `sufficient`, расположенных перед текущим, не вернул отрицательного ответа, текущий модуль вернёт положительный ответ и все последующие модули будут проигнорированы;
- `optional` — результат проверки модуля важен только в том случае, если действие является единственным для данного модуля;
- `include` — предназначается для добавления строк заданного типа из других файлов конфигурации из каталога `/etc/pam.d/` в файл конфигурации `/etc/pam.d/system-auth`. Название файла указывается в качестве аргумента действия.

Поле `module-path` задаёт путь к вызываемому модулю.

Поле `module-arguments` — дополнительные необязательные параметры модуля, необходимые для определения действий некоторых отдельных модулей в случае успешной авторизации. Так, если в конфигурационном файле найти строку, содержащую `pam_pwquality.so`, и добавить в нее `minlen=8`, то будет установлена минимальная длина пароля, равная 8-ми символам.

Пример: В качестве примера сделаем блокировку учётной записи пользователя, который совершит определенное количество неудачных попыток входа в систему.

Для этого внесем в файл `/etc/pam.d/system-auth` следующие изменения:

1. Сначала допишем в секцию `auth` строку `auth required pam_tally2.so deny=2 onerr=fail`, т.е. подключим модуль `pam_tally2` и установим блокировку пользователя после двух (значение параметра `deny`) неудачных попыток входа.
2. Затем в секции `account` добавим строку `account required pam_tally2.so` и закомментируем строки вида `auth requisite pam_succeed_if.so uid >= 1000 quiet` и `auth required pam_deny.so`.
3. Потом строку `auth sufficient pam_unix.so nullok try_first_pass` заменим на `auth required pam_unix.so nullok try_first_pass`.

После этого пользователь, допустивший подряд две неверных попытки входа, на третьей получит сообщение о том, что его учетная запись заблокирована. И даже

если четвертой попыткой он введет верный пароль, то все равно не получит доступ к системе.

```
$ sudo user2
Пароль:
sudo Сбой при проверке подлинности

$ sudo user2
Пароль:
sudo Сбой при проверке подлинности

$ sudo user2
Пароль:
Учетная запись заблокирована как следствие неудачных попыток входа (всего 3)
sudo Сбой при проверке подлинности

$ sudo user2
Пароль:
Учетная запись заблокирована как следствие неудачных попыток входа (всего 4)
sudo Сбой при проверке подлинности
```

Пример: В качестве другого примера настроим проверку паролей на сложность подбора через `pam_cracklib`.

1. Для этого добавим или изменим следующую строку:

```
password requisite pam_cracklib.so try_first_pass retry=3 type= minlen=6 dcredit=-2 ucredit=-3␣
↪lcredit=-2 ocredit=-1
```

Это значит следующее:

- после трех неуспешных попыток (`retry=3`) модуль вернет ошибку;
- минимальная длина для пароля — 6 символов (`minlen=6`);
- минимальное количество цифр — 2 (`dcredit=-2`);
- минимальное количество символов верхнего регистра — 3 (`ucredit=-3`);
- минимальное количество символов нижнего регистра — 2 (`lcredit=-2`);
- минимальное количество других символов — 1 (`ocredit=-1`).

2. Удалим или закомментируем следующую строку:

```
password requisite pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
```

Результат

- Выполним команду `passwd` для смены пароля пользователя `user2`.
- Зададим пароль из трех символов и увидим сообщение «НЕУДАЧНЫЙ ПАРОЛЬ: слишком короткий».
- Зададим пароль из четырех символов, система выдаст сообщение «НЕУДАЧНЫЙ ПАРОЛЬ: короткий».
- Зададим пароль из шести символов (букв и цифр), в результате чего получим сообщение «НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой».

- После трех неуспешных попыток модуль вернет ошибку.

```
$ passwd
Изменяется пароль пользователя user2.
Смена пароля для user2.
(текущий) пароль Unix:
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: слишком короткий
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
passwd: Использовано максимальное число попыток, заданное для службы
```

- Зададим пароль достаточной длины из одних цифр и получим сообщение «НЕУДАЧНЫЙ ПАРОЛЬ: не содержит достаточное число РАЗЛИЧНЫХ символов».

- Зададим пароль достаточной длины, содержащий все указанные требования, кроме включения в него отличных от алфавита и цифр символов. Например, `2QyfMOb4`. Получим сообщение «НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой».

```
$ passwd
Изменяется пароль пользователя user2.
Смена пароля для user2.
(текущий) пароль Unix:
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: не содержит достаточное число РАЗЛИЧНЫХ символов
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
Новый пароль:
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
passwd: Использовано максимальное число попыток, заданное для службы
```

- Зададим пароль, соблюдая все установленные требования. Например, `2QyfM*Ob4`. Пароль будет успешно задан (см. листинг).

```
$ passwd
Изменяется пароль пользователя user2.
Смена пароля для user2.
(текущий) пароль Unix:
Новый пароль:
Повторите ввод нового пароля:
passwd: Все данные аутентификации успешно обновлены.
```

Конфигурационный файл /etc/issue

Конфигурационный файл `/etc/issue` позволяет задать текстовое содержание уведомления пользователю перед началом его идентификации и аутентификации для входа в систему. Например, с предупреждением о том, что в ней реализованы меры защиты информации и о необходимости соблюдения соответствующих правил обработки данных. Традиционно в конфигурационном файле присутствуют опции выдачи сведений об операционной системе и ядре. Дополнительно можно добавить опции выдачи текущих даты и времени, количества работающих пользователей и некоторых других сведений.

Конфигурационный файл /etc/shadow

Конфигурационный файл `/etc/shadow` содержит сведения об учетных записях и паролях пользователей в виде строк со следующей структурой:

```
username:id:salt:hashed:lastchanged:min:max:warn:inactive:expire
```

Структура файла:

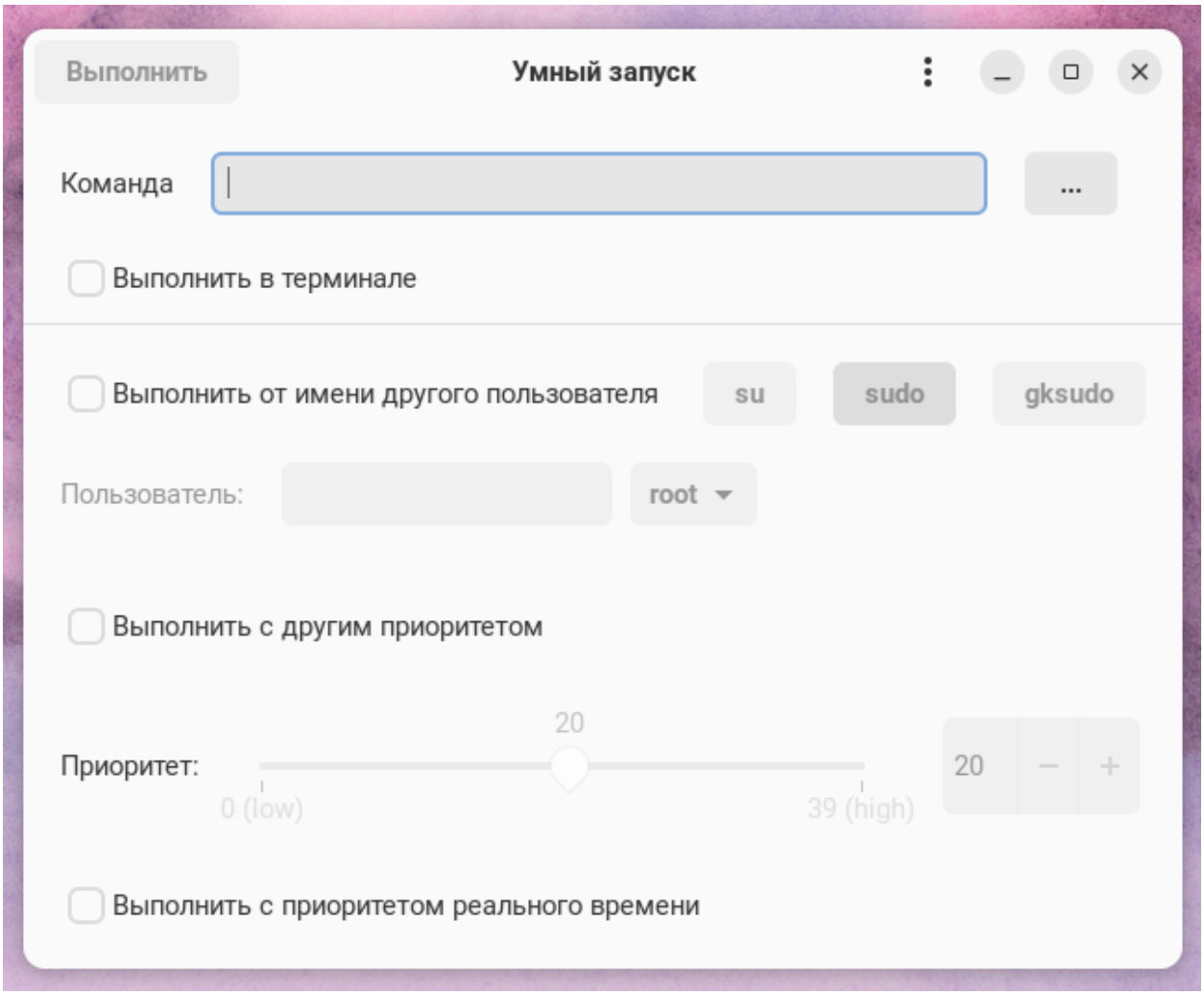
- `username` — имя пользователя:
- `id` — алгоритм шифрования: 1 (алгоритм MD5), 5 (SHA-256), 6 (SHA-512);
- `salt` — «соль», добавляемая к паролю строка из 10-20 случайных символов;
- `hashed` — зашифрованный пароль;
- `lastchanged` — дата последнего изменения пароля;
- `min` — минимальное число дней между двумя последовательными сменами паролей;
- `max` — срок действия пароля, т.е. максимальное число дней, в течение которых пароль будет активен;
- `warn` — за какое количество дней до срока истечения действия пароля пользователь будет уведомлен о том, что его необходимо сменить;
- `inactive` — количество дней после истечения срока действия пароля, спустя которое его учётная запись блокируется;
- `expire` — число дней, прошедших с момента блокирования учётной записи.

Если после имени пользователя `username` вместо `id:salt:hashed` стоит символ `*` либо последовательность из двух символов `!!`, то это означает, что попытки входа в систему от имени данного пользователя заблокированы.

Запуск программ от имени другого пользователя

Утилита smart-launcher

Утилита `smart-launcher` («Умный запуск») предназначена для запуска приложений в графическом или терминальном режиме от имени администратора системы или другого пользователя, а также для запуска приложений с изменённым приоритетом выполнения.

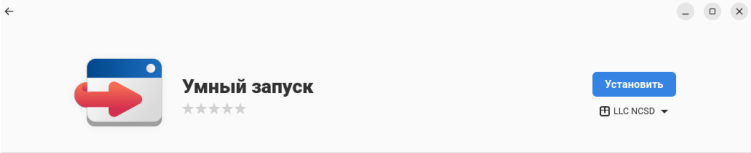


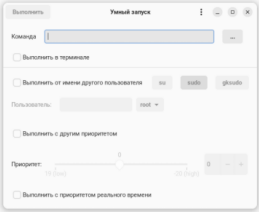
Установка и запуск

Для установки программы `smart-launcher` выполните следующую команду в «Терминале»:

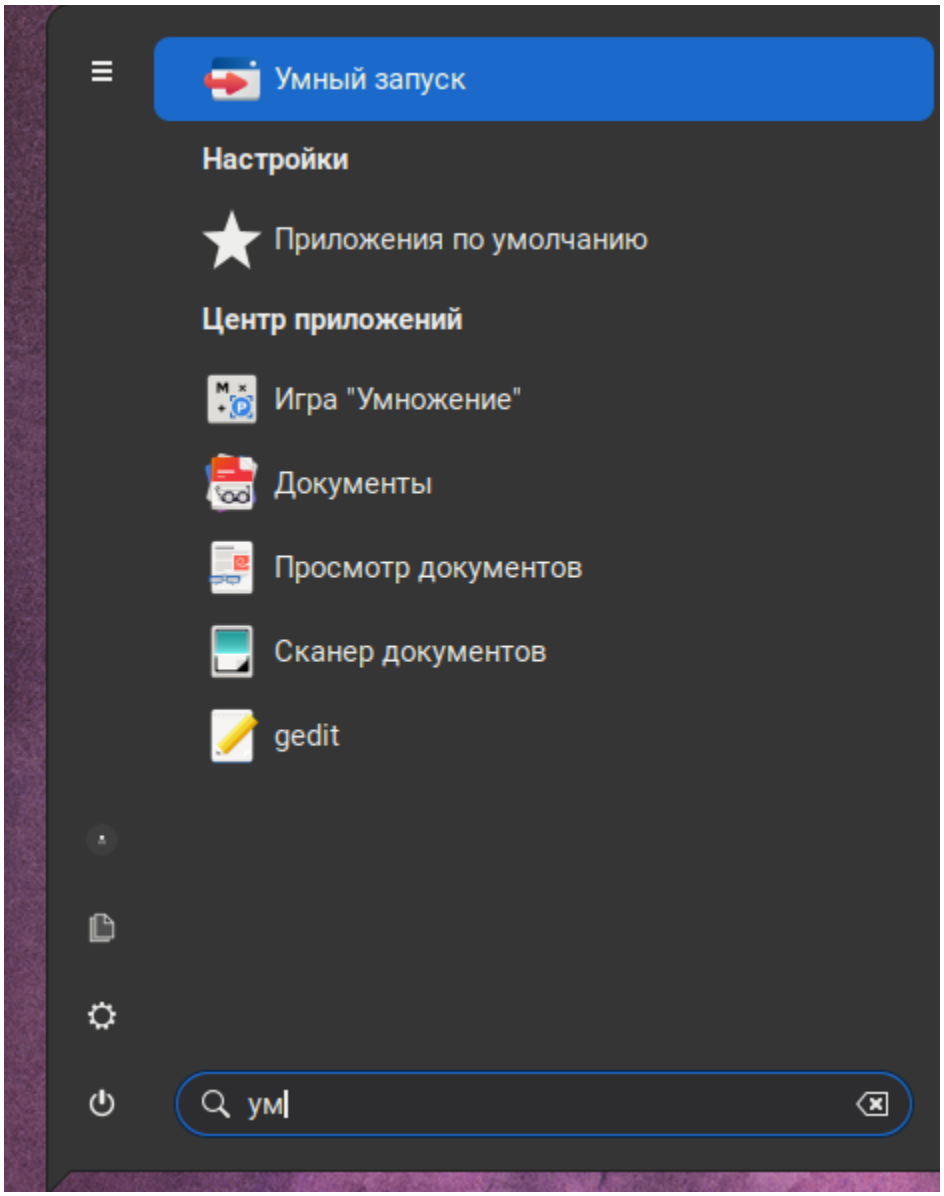
```
$ sudo dnf install smart-launcher
```

Либо установите программу через «Центр приложений».





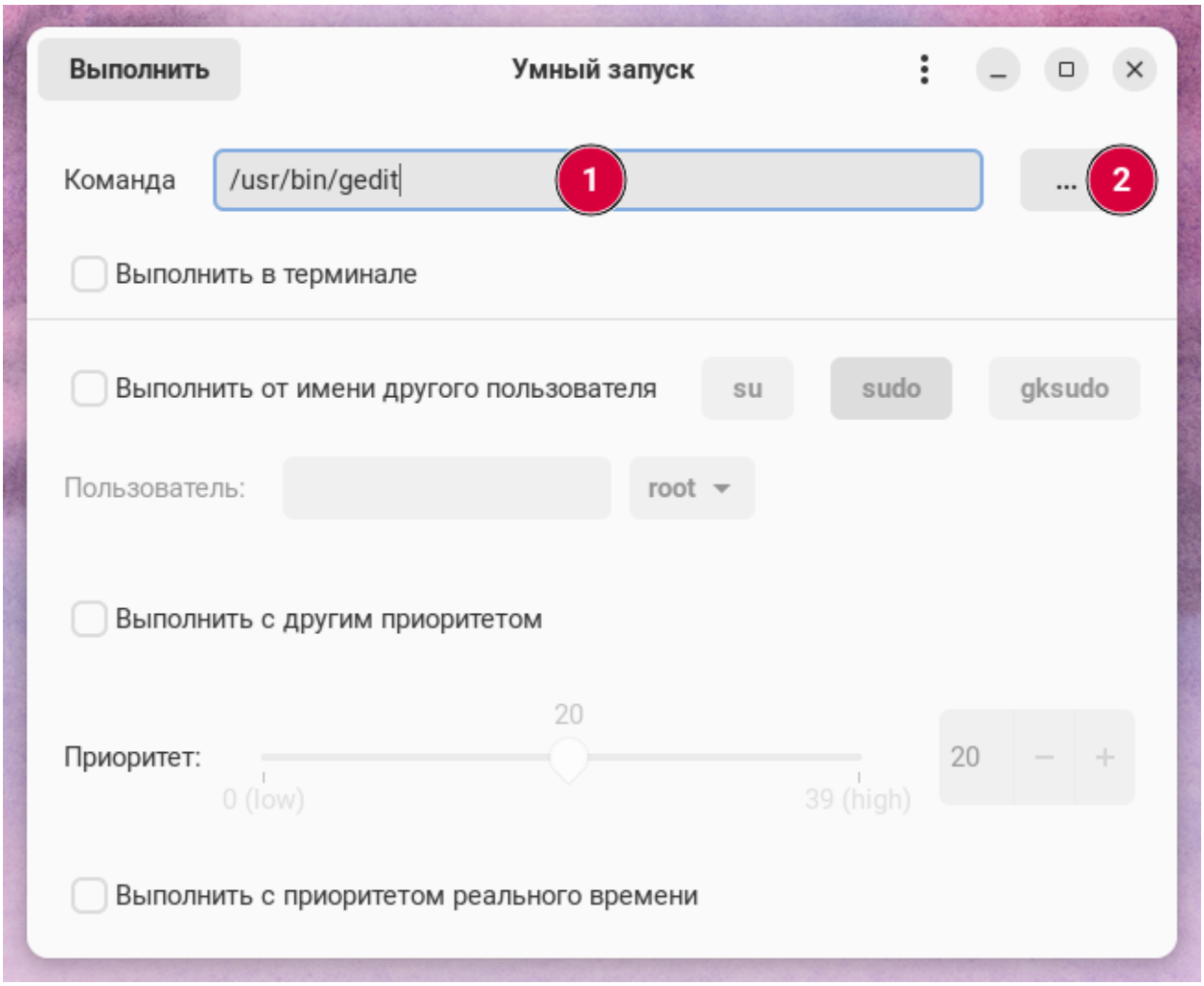
После установки `smart-launcher` можно запустить, выбрав в главном меню системы пункт «Умный запуск».



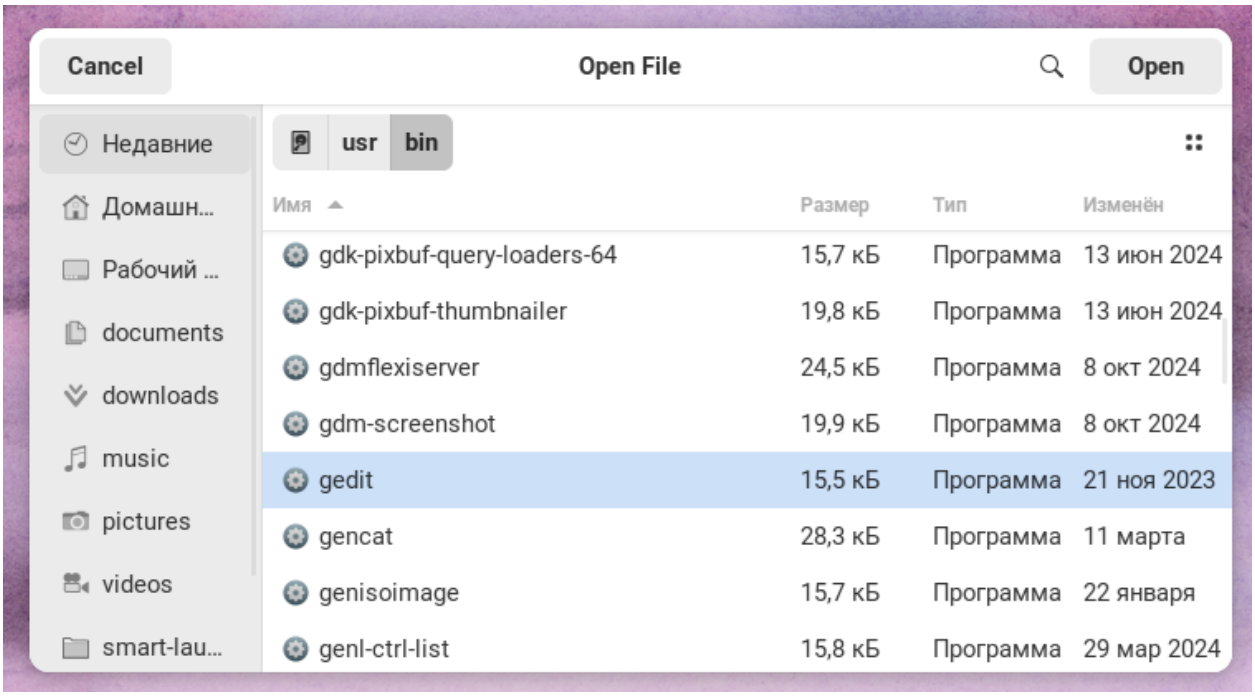
Либо запустив программу через «Терминал»:

```
$ smart-launcher
```

Для запуска графических приложений от имени другого пользователя необходимо ввести имя исполняемого файла в поле «Команда» (отмечено цифрой 1 на снимке экрана).

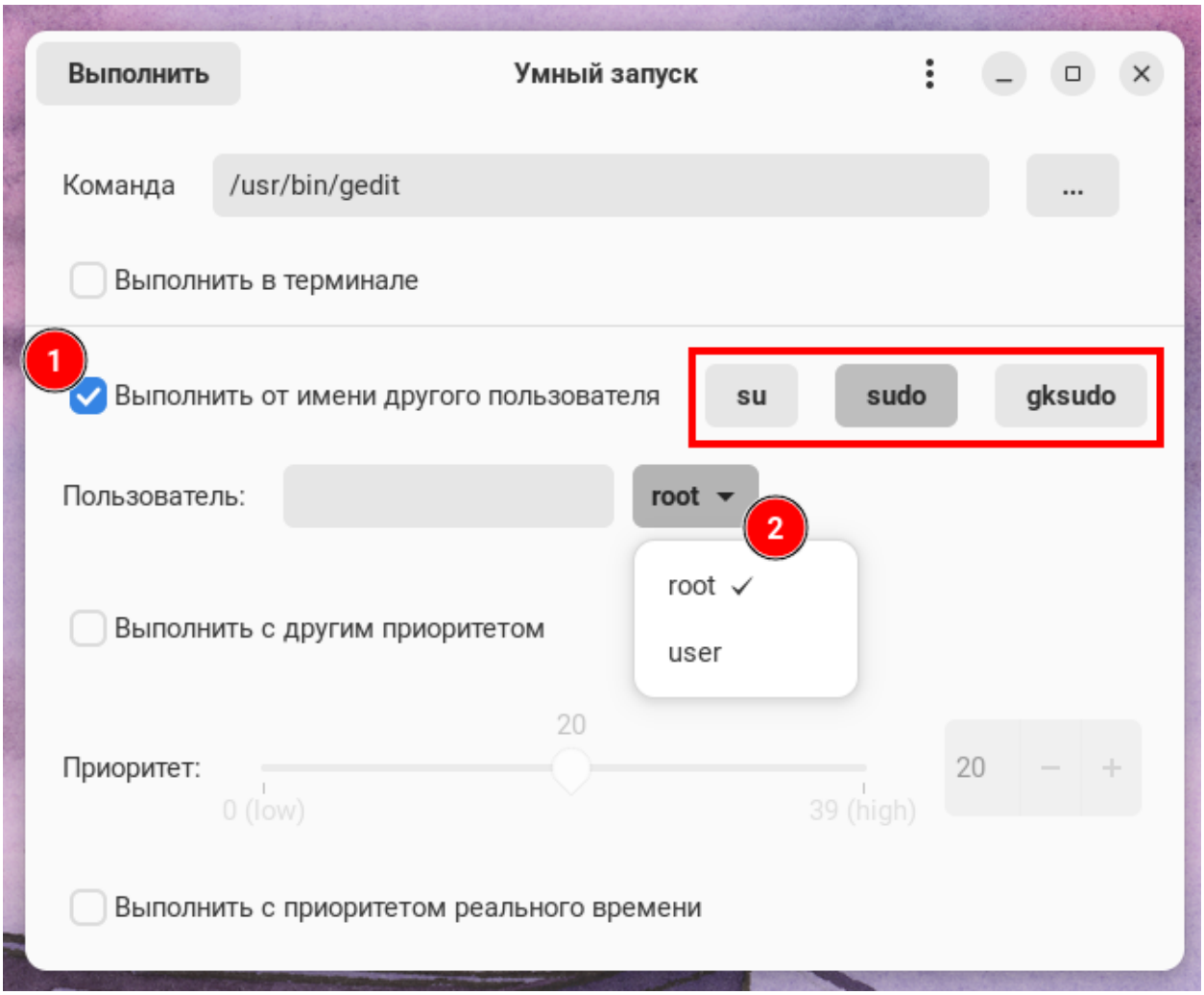


Или вызвать диалог выбора файла (кнопка «...» отмечена цифрой 2 на снимке экрана выше) и выбрать исполняемый файл.

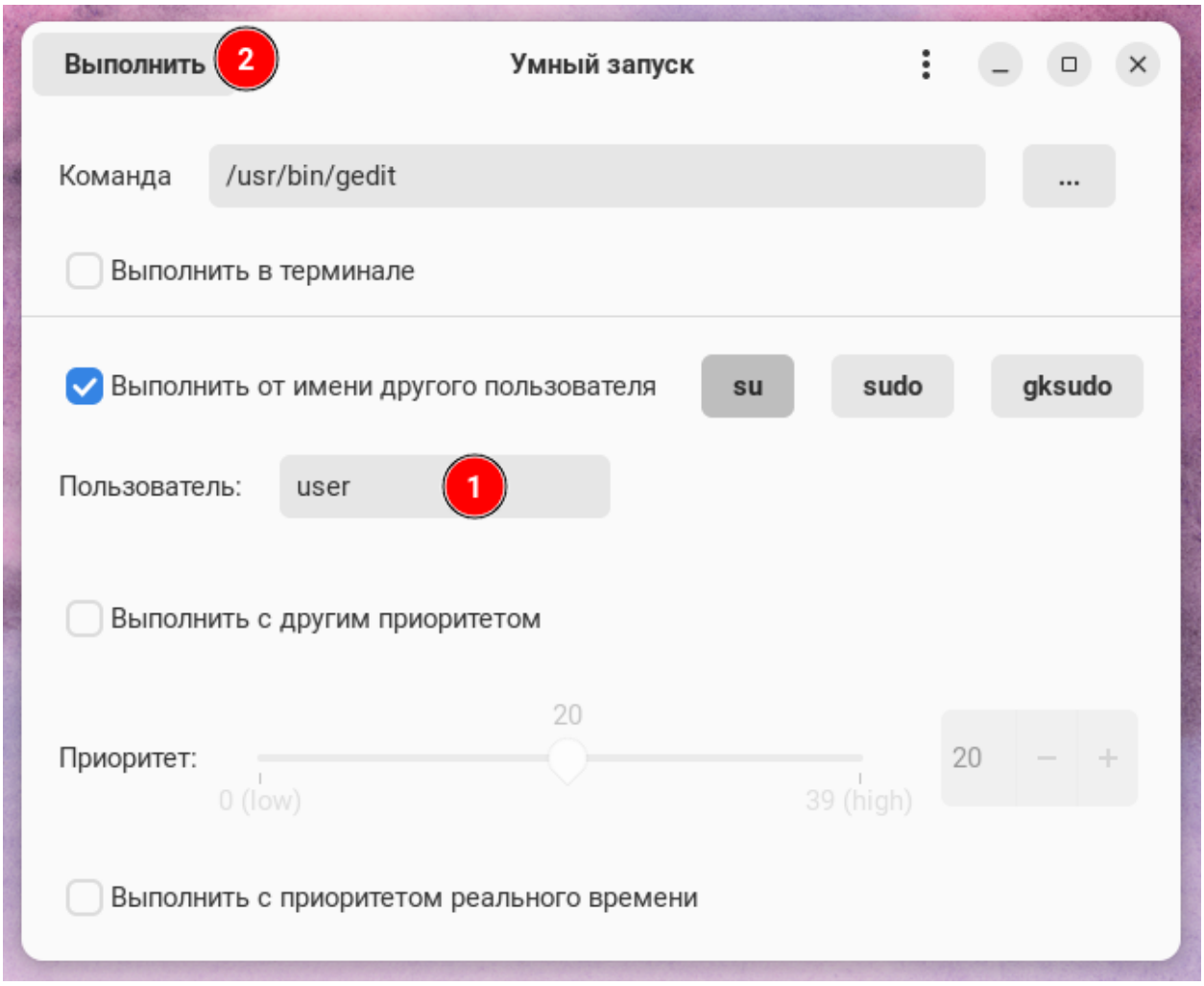


Далее, установите флажок «Выполнить от имени другого пользователя» (отмечен цифрой 1 на следующем снимке экрана) и, в случае необходимости, выберите метод запуска `su`, `sudo` (используется по умолчанию) или `gksudo` (соответствующая группа переключателей выделена красным прямоугольником на снимке экрана).

По умолчанию приложение будет выполнено от имени пользователя `root`, но вы можете выбрать другого пользователя из выпадающего списка локальных пользователей (отмечен цифрой 2 на снимке экрана).

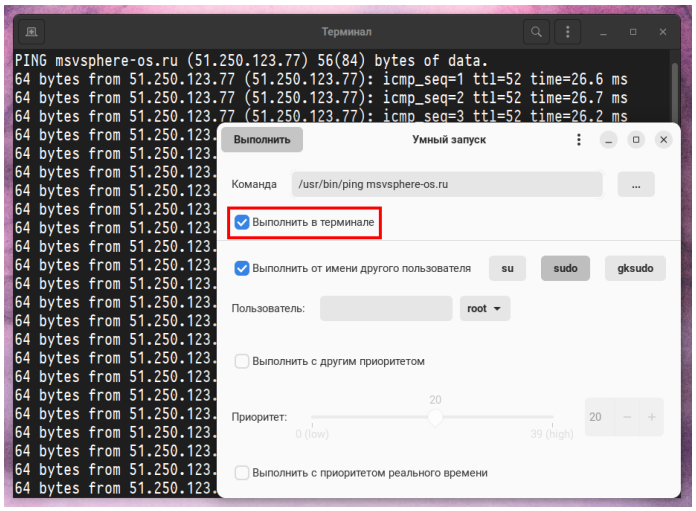


В случае, если пользователь не является локальным (например, пользователь домена Active Directory или FreeIPA), необходимо ввести его имя в поле «Пользователь» (отмечено цифрой 1 на снимке экрана).



Для запуска приложения нажмите кнопку «Выполнить» (отмечена цифрой 2 на снимке экрана выше) — приложение будет запущено в новом окне.

Для запуска приложения в терминальном режиме от имени другого пользователя необходимо выбрать исполняемый файл, при необходимости ввести имя пользователя и выбрать режим запуска. Далее, необходимо установить флажок «Выполнить в терминале» и нажать кнопку «Выполнить».



Указанная программа будет запущена в графическом терминале.

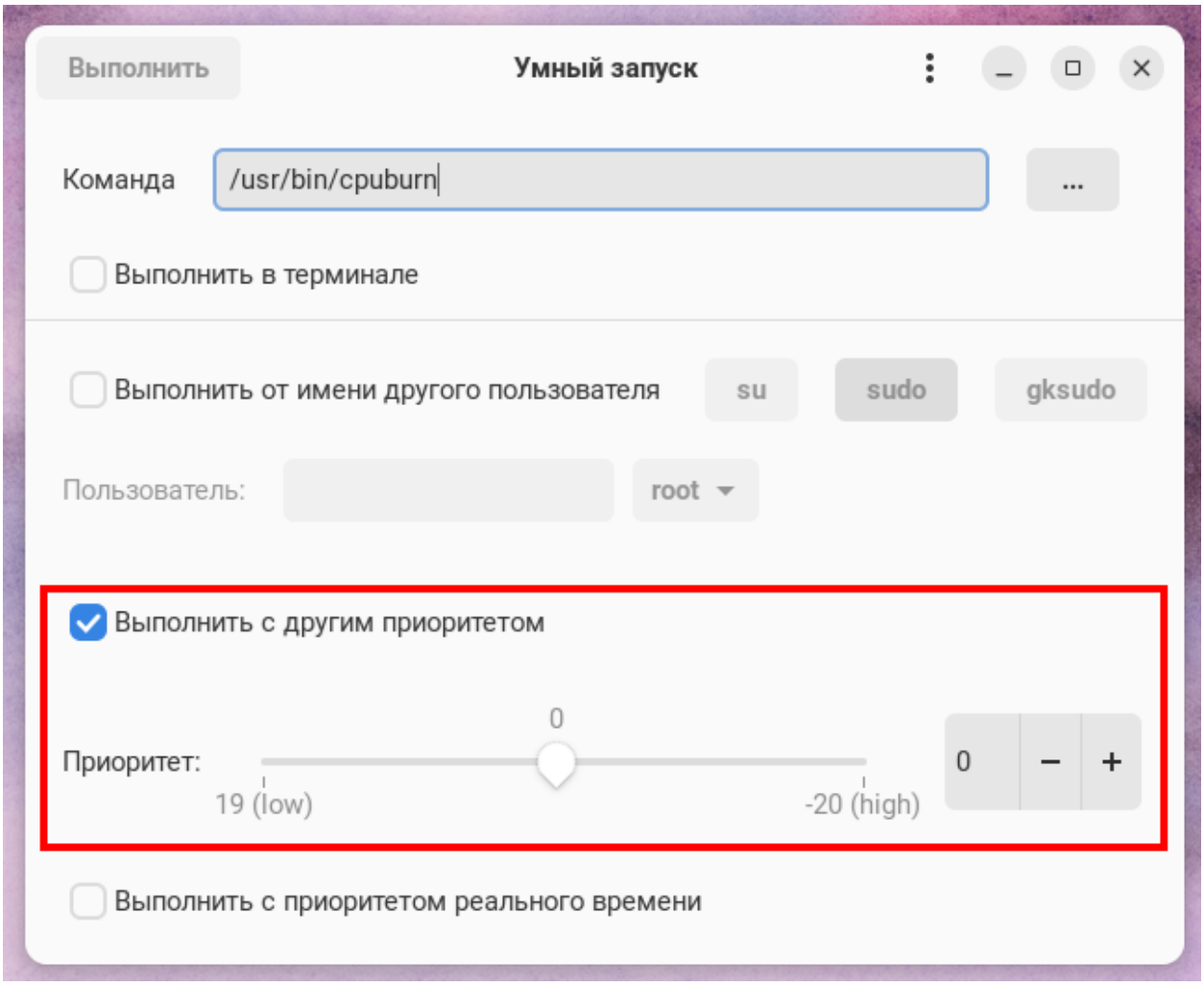
Утилита `smart-launcher` позволяет управлять двумя уровнями приоритета выполнения процесса:

- приоритет выполнения в пользовательском пространстве с использованием утилиты `nice` (см. `man nice`). В таком режиме все процессы делят ресурсы центрального процессора пропорционально установленным приоритетам;
- приоритет выполнения в реальном времени с использованием утилиты `chrt` (см. `man chrt`). Процессы, запущенные в таком режиме, имеют приоритет в реальном времени и могут вытеснять все другие процессы, в том числе и запущенные с `nice = -20`.

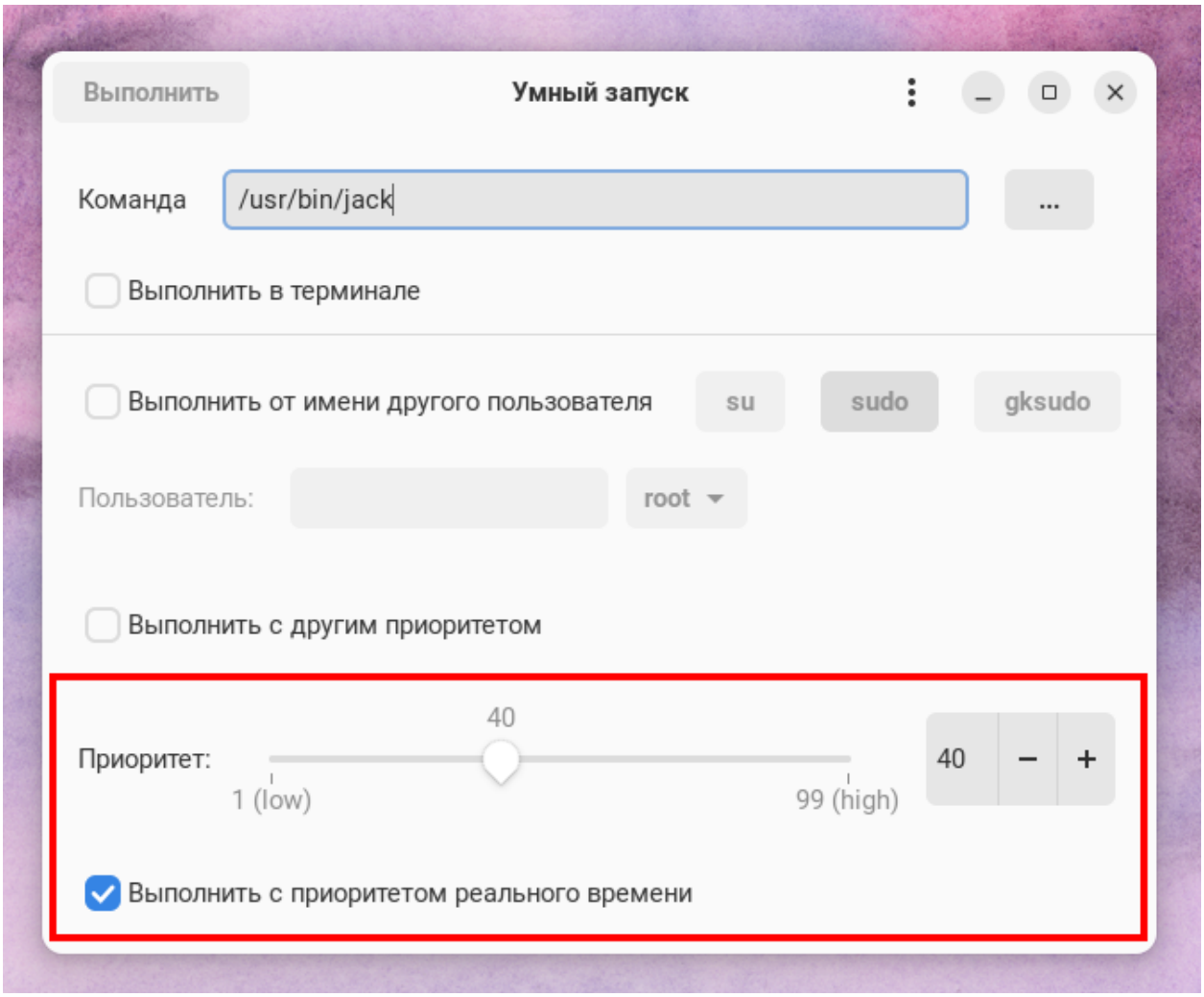
В большинстве случаев не рекомендуется изменять приоритеты реального времени за исключением тех задач, где критично время выполнения: обработка сигналов в реальном времени, промышленные контроллеры, встраиваемые системы с жёсткими требованиями к планированию выполнения процессов.

Чтобы изменить приоритет выполнения процесса вам потребуются привилегии системного администратора.

Для запуска приложения с изменённым приоритетом в пользовательском пространстве вам необходимо выбрать исполняемый файл, установить флажок «Выполнить с другим приоритетом» и установить приоритет от `19` (наименьший приоритет) до `-20` (наибольший приоритет) в поле «Приоритет» используя ползунок, кнопки `-` и `+` или указав нужное число в соответствующем поле ввода.



Для запуска приложения с приоритетом реального времени вам необходимо выбрать исполняемый файл, установить флажок «Выполнить с приоритетом реального времени» и установить приоритет от `1` (минимальный приоритет) до `99` (максимальный приоритет) в поле «Приоритет» используя ползунок, кнопки `-` и `+` или указав нужное число в соответствующем поле ввода.

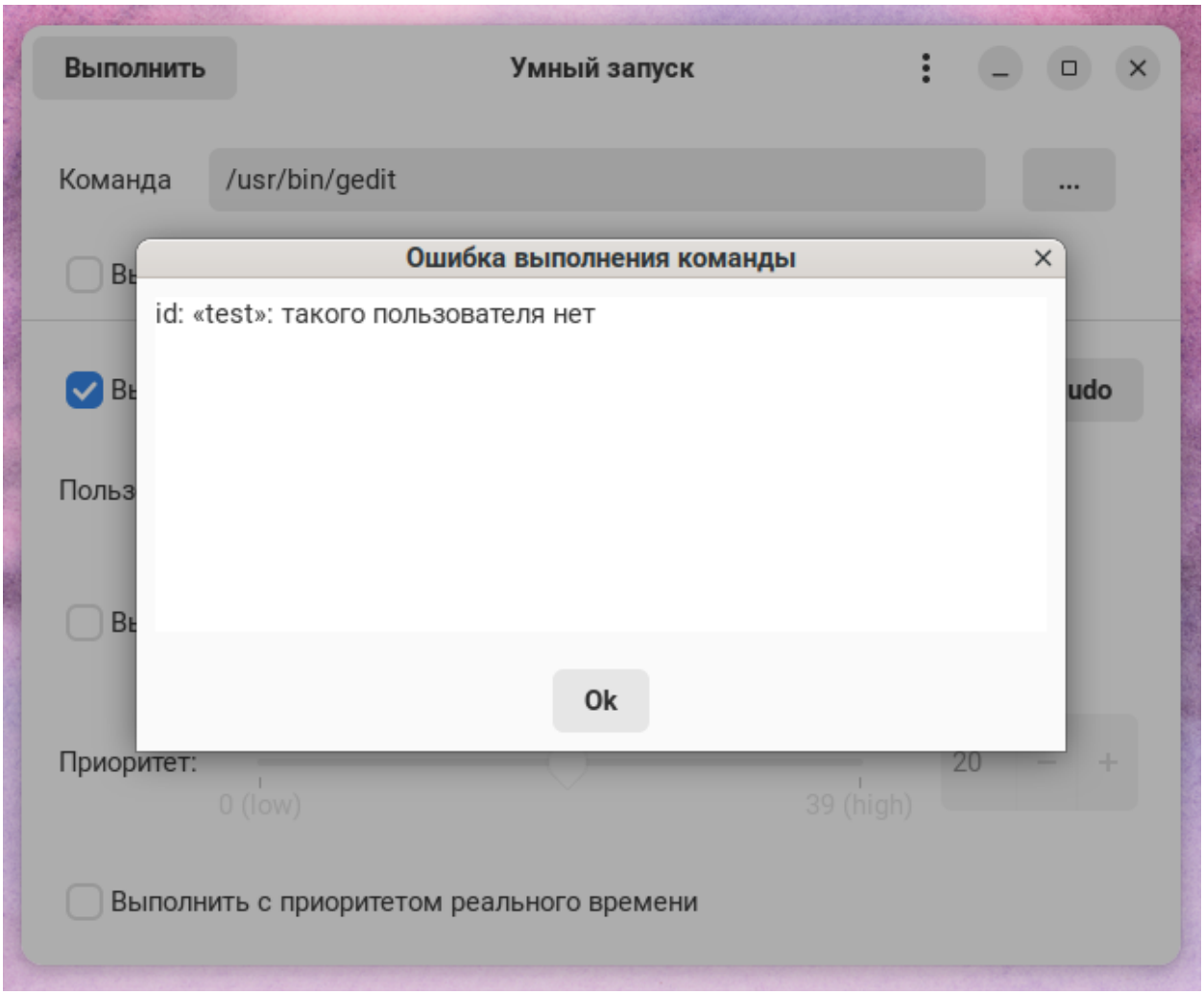


Запускать приложения с изменением уровня приоритета можно как в графическом режиме, так и в терминальном.

При запуске приложения от имени другого пользователя с использованием `sudo` или `gksudo` пароль пользователя, в случае необходимости, будет запрошен в отдельном графическом окне.

Если же приложение запускается от имени другого пользователя с использованием `su`, то пароль будет запрошен в отдельном терминальном окне.

Ошибки, возникающие во время запуска приложения, отображаются в отдельном окне.



Также `smart-launcher` регистрирует возникающие ошибки в файле журнала `~/.cache/smart-launcher/smart-launcher.log`.

Управление доступом

Введение

Средства управления доступом предоставляют возможности ограничения количества одновременно предоставляемых параллельных сеансов доступа пользователей к системе, блокирования сеанса доступа пользователя в систему после истечения установленного периода времени бездействия или по его запросу, поддержки и сохранения атрибутов безопасности, связанных с информацией в процессе её хранения и обработки, разделения полномочий пользователей и администраторов, обеспечивающих функционирование системы, реализации различных методов управления доступом, типов доступа и правил разграничения доступа, назначения приоритетов для использования субъектами доступа вычислительных ресурсов, квотирования предоставляемых вычислительных ресурсов, а также другие возможности.

Установка и изменение прав доступа к файлам и директориям

Утилита `chmod` позволяет устанавливать и изменять права доступа к файлам и директориям. Она принимает описания прав доступа в двух нотациях: численной и буквенной, описываемой ниже.

В соответствии с буквенной нотацией пользователи, которые могут потенциально работать с файлом, разделяются на владельца (`u`), группу владельцев (`g`) и всех остальных пользователей (`o`), а файл может быть читаемым (`r`), записываемым (`w`) и исполняемым (`x`).

Описание прав доступа начинается с символа, соответствующего типу пользователей. Затем идет символ `+` для установки или символ `-` для снятия прав доступа, после чего описание заканчивается последовательностью символов, соответствующей правам доступа.

Например, для определения прав доступа, позволяющих читать и модифицировать файл `file`, может использоваться следующая команда:

```
$ chmod g+rw file
```

Для удаления всех прав доступа на директорию `/directory` для группы и остальных пользователей может использоваться следующая команда:

```
$ sudo chmod go-rwx /directory.
```

Утилита поддерживает опции, перечисленные в таблице:

Таблица 12: Опции утилиты `chmod` и их значения

Опция	Значение
`-R, --recursive`	Рекурсивное изменение прав доступа для директорий и их содержимого.
`-c, --changes`	Подробно описывать действия для каждого файла, чьи права действительно изменяются.
`-f, --silent, --quiet`	Не выдавать сообщения об ошибке для файлов, чьи права не могут быть изменены.
`-v, --verbose`	Подробно описывать действие или отсутствие действия для каждого файла.

продолжение на следующей странице

Таблица 12 – продолжение с предыдущей страницы

Опция	Значение
`--version`	Сообщить информацию о версии.
`--help`	Выводит справку по этой утилите и завершает работу.

Пример: сменим права для файла `file1` так, чтобы владелец файла имел права на чтение и запись, а группа и остальные пользователи — только на чтение:

```
$ chmod u+rw g-wx o-wx file1
```

Назначение и изменение владельца файла и директории

Утилита `chown` позволяет назначить или изменить владельца файла или директории.

Утилита поддерживает следующие опции, перечисленные в таблице:

Таблица 13: Опции утилиты `chown` и их значения

Опция	Значение
`-R, --recursive`	Рекурсивное изменение прав доступа для директорий и их содержимого.
`-c, --changes`	Подробно описывать все изменения.
`-f, --silent, --quiet`	Не выдавать сообщения об ошибке.
`-v, --verbose`	Вывести подробное описание действия.
`--version`	Сообщить информацию о версии.
`--help`	Выводит справку по этой утилите и завершает работу.

Пример: назначим пользователя `user` владельцем файла `file`:

```
$ sudo chown user file
```

Пример: выполним рекурсивный обход директории `directory` и назначим пользователя `user` владельцем всех вложенных файлов:

```
$ sudo chown -R user directory
```

Изменение группы-владельца файла или директории

Утилита `chgrp` позволяет изменить группу-владельца файла или директории.

Утилита поддерживает следующие опции, перечисленные в таблице:

Таблица 14: Опции утилиты `chgrp` и их значения

Опция	Значение
`-R, --recursive`	Рекурсивное изменение группы для каталогов и всего их содержимого.
`-c, --changes`	Подробно описывать действия для каждого файла, чья группа действительно меняется.
`-f, --silent, --quiet`	Не выдавать сообщения об ошибке для файлов, чья группа не может быть изменена.
`-v, --verbose`	Подробно описывать действие или отсутствие действия для каждого файла.
`--version`	Сообщить информацию о версии.
`--help`	Вывести справку по этой утилите и завершить работу.

Пример: изменим группу-владельца файла `file` на новую группу `new_group`:

```
$ sudo chgrp new_group file
```

Просмотр и изменение списков правил контроля доступа для файлов и директорий

Утилита `setfacl` позволяет просматривать и изменять списки правил контроля доступа для файлов и директорий.

Утилита поддерживает следующие опции, перечисленные в таблице:

Таблица 15: Опции утилиты `setfacl` и их значения

Опция	Значение
`-d`	Установить правила контроля доступа по умолчанию.
`-k`	Удалить правила контроля доступа по умолчанию.
`-s`	Заменить правила контроля доступа заданными.
`-m`	Модифицировать правила контроля доступа.
`-x`	Удалить указанное правило контроля доступа.
`-b`	Удалить все правила контроля доступа.
`-v`	Вывести версию и выйти.
`-h`	Вывести справку об использовании утилиты и выйти.

Пример: удалим все правила контроля доступа к файлу `file`:

```
$ sudo setfacl -b file
```

Просмотр списков контроля доступа

Утилита `getfacl` позволяет просматривать списки контроля доступа.

Утилита поддерживает следующие опции, перечисленные в таблице:

Таблица 16: Опции утилиты `getfacl` и их значения

Опция	Значение
`-a, --access`	Выводить список контроля доступа к файлам.
`-d, --default`	Выводить список контроля доступа по умолчанию.
`-c, --omit-header`	Не выводить заголовок с комментариями.
`e, --all-effective`	Выводить комментарии с действующими правами доступа для каждого пользователя.
`-E, --no-effective`	Не выводить комментарии с действующими правами доступа ни для одного пользователя.
`-R, --recursive`	Делать рекурсивный обход директории и выводить списки контроля доступа для каждого файла и директории.
`-v, --version`	Вывести версию и выйти.
`-h, --help`	Вывести справку об использовании утилиты и выйти.

Пример: просмотрим список контроля доступа для файла `cg.conf`:

```
$ getfacl cg.conf
# file: cg.conf
# owner: user
# group: user
user::rwx
group::r-x
other::r-x
```

Пример: зададим дополнительные компоненты списка контроля доступа для пользователя `user` и группы `user` по отношению к файлу `cg.conf`:

```
$ setfacl -m g:user:rxw cg.conf

$ setfacl -m u:user:rxw cg.conf

$ getfacl cg.conf
# file: cg.conf
# owner: user
# group: user
user::rwx
user:user:rwx
group::r-x
group:user:rwx
mask::rwx
other::r-x
```

Пример: от имени администратора модифицируем списки контроля доступа для файлов, владельцем которых он является:

```
$ sudo setfacl -m u:user:rxw ~/file2

$ sudo getfacl ~/file2
getfacl: Removing leading `/` from absolute path names
# file: root/file2
# owner: root
# group: root
user::rw-
user:user:rwx
group::r--
mask::rwx
other::r--
```

Пример: от имени администратора модифицируем списки контроля доступа для файлов, владельцем которых он не является:

```
$ sudo setfacl -m u:user:rxw /home/user3/file2

$ sudo setfacl -m u:user:rxw /home/user3/dir2

$ sudo getfacl /home/user3/file2
getfacl: Removing leading `/` from absolute path names
# file: home/user3/file2
# owner: user3
# group: user3
user::rwx
user:user:rwx
group::---
mask::rwx
other::---

$ sudo getfacl /home/user3/dir2
getfacl: Removing leading `/` from absolute path names
# file: home/user3/dir2/
# owner: user3
# group: user3
user::rwx
user:user:rwx
group::---
mask::rwx
other::---
```

Пример: от имени администратора удалим списки контроля доступа для объектов, владельцем которых он является:

```
$ sudo setfacl -b ~/file2

$ sudo getfacl ~/file2
getfacl: Removing leading `/` from absolute path names
# file: root/file2
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Пример: от имени администратора удалим списки контроля доступа для объектов, владельцем которых он не является:

```
$ sudo setfacl -b /home/user3/file2

$ sudo getfacl /home/user3/file2
getfacl: Removing leading `/` from absolute path names
# file: home/user3/file2
# owner: user3
# group: user3
user::rwx
group::---
other::---
```

Важно

Пользователь, не обладающий полномочиями администратора, не может удалять списки контроля доступа, которые он не создавал.

Редактирование пользовательских квот для файловой системы

Утилита `edquota` позволяет редактировать пользовательские квоты для файловой системы.

Утилита поддерживает следующие опции, перечисленные в таблице:

Таблица 17: Опции утилиты `edquota` и их значения

Опция	Значение
`-u, --user`	Изменить пользовательскую квоту.
`-g, --group`	Изменить групповую квоту.
`-p, --prototype = protoname`	Дублировать квоты прототипного пользователя. Это обычный механизм, используемый для инициализации квот для групп пользователей.
`-F, --format = имя-формата`	Изменить квоту для указанного формата.
`-f, --filesystem`	Выполнять указанные операции только для заданной файловой системы. По умолчанию операция выполняется для всех файловых систем с квотой.
`-t, --edit-period`	Редактировать мягкие ограничения по времени для каждой файловой системы.
`-T, --edit-times`	Изменить время для пользователя или группы, когда принудительное ограничение установлено.

Конфигурационный файл /etc/profile

Конфигурационный файл `/etc/profile` используется для задания элементов окружения оболочки пользователя. Например, в нём определяются глобальные переменные:

- `PATH` — переменная среды, используемая для указания оболочке списка каталогов, которые будут просматриваться при поиске исполняемых файлов;
- `USER` — имя пользователя при входе в ОС;
- `LOGNAME` — то же, что и `USER`. Некоторые программы считывают значение этой глобальной переменной вместо `USER`;
- `MAIL` — имя файла, в который записывается локальная почта пользователя, а также его расположение;
- `HOSTNAME` — имя хоста;
- `HISTSIZE` — количество исполненных команд, сохраняемых в истории;
- `HISTCONTROL` — политики в отношении команд, сохраняемых в истории. По умолчанию задано значение `ignoredups`, то есть команда, полностью совпадающая с одной из уже записанных в историю, не сохраняется. Если задать политику `ignorespace`, то будут игнорироваться как дублирующиеся команды, так и те, что начинаются с символа пробела.

Также в конфигурационном файле задаётся маска, используемая для определения конечных прав доступа для пользователя.

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`/usr/bin/id -u`
        UID=`/usr/bin/id -ru`
    fi
    USER="`/usr/bin/id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
fi
```

Пример: определим время бездействия при локальной терминальной сессии равным двум минутам (120 с). Для этого в файле `/etc/profile` после строк

```
HOSTNAME= '/usr/bin/hostname 2>/dev/null'
HISTSIZE=1000
```

Добавим строку `TMOUT=120`. Там же, в строке

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

Необходимо добавить параметр `TMOUT`:

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE TMOUT HISTCONTROL
```

Для подтверждения вступления изменений в силу надо будет завершить сеанс и зарегистрироваться в системе заново. Тогда появится сообщение, что после двух минут бездействия время ожидания ввода вышло, в результате чего интерактивный сеанс был закрыт.

Конфигурационный файл /etc/security/limits.conf

Конфигурационный файл `/etc/security/limits.conf` может использоваться для задания модулю `pam_limits.so` дополнительных ограничений. Для этого каждая его строка включает четыре группы параметров, которые перечислены и описаны ниже:

Важно

По умолчанию все ограничения отключены — все строки закомментированы.

- **`<domain>`:**
  имя пользователя, имя группы с синтаксисом `@group`, подстановочный знак * для записи по умолчанию, подстановочный знак `%`, который также может использоваться с синтаксисом `%group` для ограничения `maxlogin`;
- **`<type>`:**
  - `soft` для установки мягких ограничений;
  - `hard` для установки жестких ограничений.
- **`<item>`:**
  - `core`: ограничивает размер файла ядра в Кб;
  - `data`: максимальный размер данных в Кб;
  - `fsize`: максимальный размер файла в Кб;
  - `memlock`: максимальное адресное пространство, предустановленное в памяти, в Кб;
  - `nofile`: максимальное количество открытых файлов;
  - `rss`: максимальный размер резидентного набора в Кб;
  - `stack`: максимальный размер стека в Кб;

- `cpu`: максимальное время процессора в MIN;
- `nproc`: максимальное количество процессов;
- `as`: ограничение адресного пространства в Кб;
- `maxlogins`: максимальное количество логинов для этого пользователя;
- `maxsyslogins`: максимальное количество входов в систему;
- `priority`: приоритет процессов пользователя;
- `locks`: максимальное количество блокировок файлов, которое может быть обеспечено пользователем;
- `sigpending`: максимальное количество ожидающих сигналов;
- `msgqueue`: максимальный объем памяти, используемый очередями сообщений POSIX, в байтах;
- `nice`: приоритет для запуска процессов утилитой `nice`;
- `rtprio`: максимальный приоритет в реальном времени.

```
# /etc/security/limits.conf
#
#This file sets the resource limits for the users logged in via PAM.
#It does not affect resource limits of the system services.
#
#Also note that configuration files in /etc/security/limits.d directory,
#which are read in alphabetical order, override the settings in this
#file in case the domain is the same or more specific.
#That means, for example, that setting a limit for wildcard domain here
#can be overridden with a wildcard setting in a config file in the
#subdirectory, but a user specific setting here can be overridden only
#with a user specific setting in the subdirectory.
#
#Each line describes a limit for a user in the form:
#
#<domain>        <type>  <item>  <value>
#
#Where:
#<domain> can be:
#        - a user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
#        - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open file descriptors
```

Пример: ограничим число параллельных сеансов доступа для каждой учетной записи пользователя. Для этого добавим в конфигурационный файл строку следующего содержания:

```
username hard maxlogins 2
```

Тогда, при условии, что пользователь `username` открыл локальную сессию (учитывая, что при входе в графический сеанс открываются сразу две сессии пользователя) и попытался зайти в систему через ssh-соединение (потенциально ещё один активный сеанс), ему будет выведено сообщение `Too many logins for 'username'` и это соединение будет заблокировано.

Конфигурационный файл /etc/fstab

Конфигурационный файл `/etc/fstab` используется для настройки параметров монтирования различных блочных устройств, разделов на диске и файловых систем. Он состоит из набора так называемых определений, каждое из которых занимает свою строку и состоит из шести полей, разделённых пробелами или символами табуляции:

```
fs_spec fs_file fs_vfstype fs_mntops fs_freq fs_passno
```

Поля предназначены для задания следующих параметров:

- **`fs_spec`**
  Физическое размещение файловой системы, по которому определяется конкретный раздел или устройство хранения для монтирования. Вместо указания размещения файловой системы явным образом можно воспользоваться её уникальным идентификатором UUID.

- **`fs_file`**
  Точка монтирования, куда монтируется корень файловой системы.

- **`fs_vfstype`**
  Тип файловой системы. Поддерживаются следующие типы: adfs, affs, autofs, coda, coherent, cramfs, devpts, efs, ext2, ext3, ext4, hfs, hpfs, iso9660, jfs, minix, msdos, nvpfs, nfs, ntfs, proc, qnx4, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, vfat, xenix, xfs.

- **`fs_mntops`**
  Опции монтирования файловой системы. Основные опции: `defaults`, `noauto`, `user`, `owner`, `comment`, `nofail`.

- **`fs_freq`**
  Предназначено для использования утилитой создания резервных копий в файловой системе. Возможные значения: `0` и `1`. Если указано `1`, то утилита создаст резервную копию.

- **`fs_passno`**
  Предназначено для использования программой `fsck` при необходимости проверки целостности файловой системы; возможные значения: `0`, `1` и `2`. Значение `1` указывается только для корневой файловой системы (то есть файловой системы с точкой монтирования `/`). Для остальных файловых систем для проверки утилитой `fsck` задаётся значение `2`. При значении `0` — проверка выполняться не будет.

По умолчанию конфигурационный файл включает:

```
/dev/mapper/MSVSphere-root / xfs defaults 0 0
```

Файловая система `/dev/mapper/MSVSphere-root` примонтирована в каталог `/`, тип файловой системы — `xfs`, используемые опции — `defaults`, резервная копия данных

не создаётся (`fs_freq=0`), проверка целостности файловой системы не выполняется (`fs_passno=0`).

```
UUID=b1bfe9b0-96ea-4876-883c-a9f1b6c74b /boot ext4 defaults 1 2
```

Файловая система с идентификатором `b1bfe9b0-96ea-4876-883c-a9f1b6c74b` смонтирована в `/boot`, тип файловой системы — `ext4`, используемые опции — `defaults`, резервная копия данных создаётся (`fs_freq=1`), проверка целостности файловой системы выполняется (`fs_passno=2`).

```
/dev/mapper/MSVSphere-swap swap defaults 0 0
```

Файловая система `/dev/mapper/MSVSphere-swap` является разделом подкачки `swap`, используемые опции — `defaults`, резервная копия данных не создаётся (`fs_freq=0`), проверка целостности файловой системы не выполняется (`fs_passno=0`).

```
# /etc/fstab
# Created by anaconda on Tue Jun 20 11:58:05 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/msvsphere-root /                       xfs     defaults        0 0
UUID=8e41c721-164e-455c-bd65-b60ad5ad7cb4 /boot                   xfs     defaults
```

Регистрация событий безопасности

Введение

В операционную систему МСВСфера 9 встроена подсистема аудита, основной задачей которой является регистрация и обработка событий, связанных с нарушением безопасности. Фильтрация сообщений происходит на основе предварительно настроенных правил, отфильтрованные события регистрируются в системном журнале событий безопасности. Системному администратору доступны различные инструменты, упрощающие анализ зарегистрированных событий.

Ниже приведены некоторые сценарии использования подсистемы аудита.

- **Регистрация событий входа в систему**

  Позволяет отслеживать как успешные, так и неудачные попытки входа в систему с использованием различных механизмов аутентификации, таких как локальная база пользователей или LDAP-каталог, SSH, Kerberos и т.п..

- **Отслеживание доступа к файлам**

  Позволяет отслеживать, осуществлялся ли доступ к тому или иному файлу или каталогу, их модификация, изменение атрибутов или запуск.

- **Мониторинг системных вызовов**

  С помощью подсистемы аудита можно отслеживать использование отдельных системных вызовов (`man 2 syscalls`), например, операции монтирования файловых систем, изменение системного времени, открытие сетевого соединения и т.п..

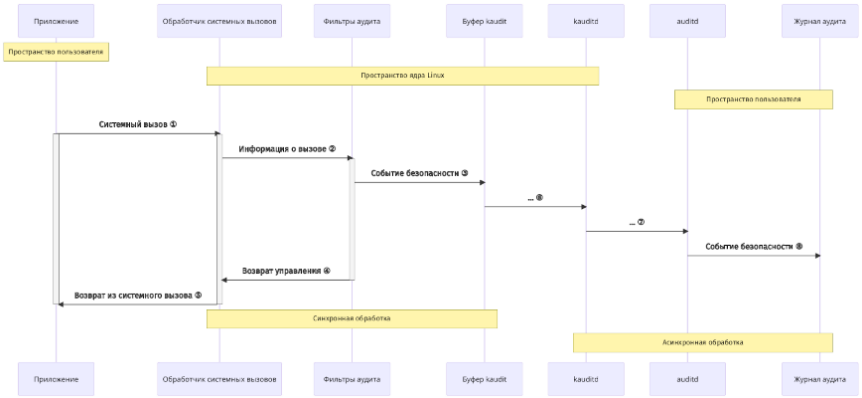
- **Отслеживание событий безопасности средства виртуализации**

  Позволяет отслеживать различные действия пользователей с ресурсами, управляемыми гипервизором `libvirt`.

- **Регистрация событий установки или удаления ПО**

  С помощью подсистемы аудита можно отслеживать установку, удаление или обновление пакетов с использованием различных пакетных менеджеров: dnf/yum, pip, npm, cpan, gem и т.д..

Рассмотрим основные компоненты и архитектуру подсистемы аудита подробнее.



1. Когда программа выполняет системный вызов, информация об этом попадает в специальный обработчик в ядре Linux — на этом этапе формируется событие безопасности, которое содержит информацию о выполняемом системном вызове, вызвавшем его процессе, идентификатор пользователя и т.д..
2. Из обработчика событие направляется на фильтрацию в соответствующий блок подсистемы аудита, где к этому событию применяются заранее загруженные фильтры.
3. В случае срабатывания одного из фильтров событие направляется в буфер событий компонента `kauditd`, ответственного за регистрацию на стороне ядра.
4. После записи события в буфер `kauditd` осуществляется возврат управления в обработчик системного вызова.
5. Из обработчика системного вызова осуществляется возврат управления в программу.
6. Из буфера событие попадает на обработку в компонент `kauditd`.
7. Компонент `kauditd` передаёт информацию о событии из пространства ядра в сервис `auditd`, работающий в пользовательском пространстве, с помощью стандартного механизма коммуникации `netlink` (`man 7 netlink`).
8. Сервис `auditd` регистрирует событие безопасности в файле системного журнала (по умолчанию в `/var/log/audit/audit.log`).

После этого регистрация события безопасности считается выполненной. Далее возможны следующие варианты обработки события безопасности в пользовательском пространстве.

- Сервис `auditd` может передать информацию о событии безопасности на обработку в другие сервисы/программы с помощью расширений (плагинов). Получателями такой информации могут быть системы централизованного сбора и анализа системных журналов, системы мониторинга, накладные средства защиты информации (СЗИ) и т.п..

- Системный администратор может выполнять анализ и обработку событий безопасности из системного журнала с помощью встроенных средств, таких как `ausearch`, `aureport`, `aulastlog` и других инструментов.

Настройка сервиса auditd

Конфигурационный файл /etc/audit/auditd.conf

Настройка сервиса `auditd` осуществляется через конфигурационный файл `/etc/audit/auditd.conf`, использующий стандартный для Unix-подобных систем формат `КЛЮЧ = ЗНАЧЕНИЕ`. Все ключи и значения являются регистронезависимыми.

В таблице ниже приведено описание допустимых параметров и их значения по умолчанию.

Таблица 18: Параметры сервиса `auditd` и их значения по умолчанию

Параметр	*Значение по умолчанию*	*Описание*
`local_events`	`yes`	Включает (`yes`) или выключает (`no`) регистрацию событий безопасности локальной системы. Если необходимо только регистрировать события, полученные по сети, установите в значение `no`. Обычно это используется при развёртывании `auditd` в контейнере для централизованного сбора информации с нескольких систем.
`log_file`	`/var/log/audit/audit.log`	Полный путь к файлу, в который необходимо записывать журнал событий безопасности.
`write_logs`	`yes`	Включает (`yes`) или выключает (`no`) запись журналов безопасности на диск.
`log_format`	`ENRICHED`	Определяет, в каком формате будет сохраняться информация в журнал. Допустимые значения параметра перечислены после таблицы.
`log_group`	`root`	Системная группа, на которую распространяются права на файлы журнала событий безопасности. Допускается использование как идентификатора группы, так и её имени.
`priority_boost`	`4`	Неотрицательное число, определяющее приоритет выполнения службы аудита. Чтобы оставить приоритет без изменений, используйте значение `0`.
`krb5_principal`	`auditd`	Имя Kerberos-принципала сервера `auditd`. При использовании значения по умолчанию сервер `auditd` будет искать ключ с именем `auditd/HOSTNAME@REALM` в файле, заданном директивой `krb5_key_file`, где `HOSTNAME` — это каноническое имя сервера, возвращаемое службой DNS по его IP-адресу, а `REALM` — область (realm) Kerberos.
`freq`	`50`	Неотрицательное число, которое определяет сколько событий необходимо записать прежде чем выполнить принудительную синхронизацию данных на диск. Этот параметр применяется только если значение `flush` установлено в `INCREMENTAL` или `INCREMENTAL_ASYNC`.
`name_format`	`NONE`	Определяет как имена компьютеров вставляются в поток событий безопасности. Допустимые значения параметра перечислены после таблицы.
`name`	Не задано	Строка, определяемая системным администратором, для использования в качестве имени системы если параметр `name_format` установлен в значение `USER`.
`max_log_file`	`8`	Максимальный размер файла журналов в мегабайтах. При достижении этого лимита будет запускаться действие, настраиваемое с помощью опции `max_log_file_action`.
`max_log_file_action`	`ROTATE`	Указывает какое действие необходимо предпринять когда система обнаружит, что достигнут лимит на максимальный размер файла журнала. Допустимые значения параметра перечислены после таблицы.
`verify_email`	`yes`	Если установлен в `yes`, то для доменного имени, указанного в почтовом адресе в директиве `action_mail_acct`, будет выполнена проверка на наличие соответствующей DNS-записи. Этот параметр должен быть расположен перед опцией `action_mail_acct` в конфигурационном файле, иначе будет использовано значение по умолчанию `yes`.
`flush`	`INCREMENTAL_ASYNC`	Определяет стратегию работы с дисковым буфером. Допустимые значения параметра перечислены после таблицы.
`q_depth`	`2000`	Определяет максимальный размер внутренней очереди диспетчера событий `auditd`. Очередь большего размера позволяет службе лучше справляться с большим потоком событий, но при слишком большой очереди некоторые события могут не успеть быть обработанными при завершении работы сервиса. Если в системном журнале появляются сообщения о том, что некоторые события были удалены, то увеличьте это значение.
`num_logs`	`5`	Определяет максимальное количество файлов журналов, которые необходимо сохранять, если параметр `max_log_file_action` установлен в значение `ROTATE`. Если значение `num_logs` меньше `2`, то ротация файлов журналов не будет производиться. В качестве значения допустимо любое число в диапазоне от `0` до `999`. С увеличением количества сохраняемых файлов может потребоваться поднять лимит на максимальное количество ожидающих запросов к сервису `auditd` — за это отвечает опция `-b` в конфигурационном файле `/etc/audit/audit.rules`. Если настроена ротация журналов, то `auditd` будет следить за количеством файлов журналов и удалять лишние файлы. Проверка избыточных журналов выполняется только при запуске сервиса и при проверке изменения конфигурации сервиса.
`space_left_action`	`SYSLOG`	Определяет какое действие необходимо предпринять если на файловой системе начинает заканчиваться свободное пространство. Допустимые значения параметра перечислены после таблицы.

продолжение на следующей странице

Таблица 18 – продолжение с предыдущей страницы

Параметр	*Значение по умолчанию*	*Описание*
action_mail_acct	root	Адрес электронной почты или псевдоним (см. /etc/aliases), на который будут отправляться сообщения от auditd. Если адрес не является локальным для данной системы, то необходимо чтобы на ней была настроена почтовая система для отправки, в том числе, требуется наличие программы /usr/lib/sendmail, предоставляемой postfix, exim, sendmail и другими почтовыми системами.
space_left	75	Если объём свободного места в файловой системе, содержащей log_file, становится меньше указанного значения, то сервис auditd выполнит действие, определённое директивой space_left_action. Если значение указано как целое число, то оно интерпретируется как абсолютный размер в мегабайтах. Если значение указано в виде числа от 1 до 99 со знаком процента (например, 5%), то auditd самостоятельно вычислит соответствующее значение как указанный процент от общего размера файловой системы.
use_libwrap	yes	Определяет, следует ли использовать механизм tcp_wrappers для фильтрации подключений от других компьютеров. Допустимые значения: yes или no.
admin_space_left	50	Если объём свободного места в файловой системе, содержащей log_file, становится меньше указанного значения, то сервис auditd выполнит действие, определённое директивой admin_space_left_action. Как и для директивы space_left, значение указывается либо в виде целого числа (абсолютный размер в мегабайтах), либо в процентах от 1% до 99%. В любом случае, значение admin_space_left должно быть меньше space_left — это следует рассматривать как последний шанс что-то предпринять прежде чем дисковое пространство будет полностью заполнено.
admin_space-_left_action	SUSPEND	Определяет какое действие необходимо предпринять если на файловой системе почти закончилось место. Список допустимых значений и их поведение совпадает с директивой space_left_action: IGNORE, SYSLOG, ROTATE, EMAIL, EXEC /путь-к-программе, SUSPEND, SINGLE и HALT.
max_restarts	10	Неотрицательное число, которое определяет сколько раз служба auditd может попытаться перезапустить вышедшее из строя расширение (плагин).
disk_full_action	SUSPEND	Определяет какое действие необходимо предпринять если на файловой системе закончилось место. Допустимые значения: IGNORE, SYSLOG, ROTATE, EXEC /путь-к-программе, SUSPEND, SINGLE, HALT. Поведение для каждого из значений рассмотрено в описании к директиве space_left_action.
disk_error_action	SUSPEND	Определяет какое действие необходимо предпринять в случае возникновения ошибки при сохранении событий безопасности на диск или при ротации файлов журнала. Допустимые значения параметра перечислены после таблицы.
tcp_listen_port	Не задано	Числовое значение от 1 до 65535, при указании которого сервис auditd будет принимать записи о событиях безопасности от удалённых систем на соответствующем TCP-порту. Для работы с удалённым сервером или клиентами рекомендуется настроить сервис auditd, чтобы он запускался после активации сетевых интерфейсов, соответствующая инструкция приведена в файле /usr/lib/systemd/system/auditd.service.
tcp_listen_queue	5	Определяет максимально разрешённое количество ожидающих (запрошенных, но ещё не принятых) сетевых подключений к сервису auditd. Слишком маленькое значение может привести к тому, что некоторые подключения будут отклонены в случае если множество клиентов будет подключаться одновременно, допустим, после сбоя питания. Эта опция используется только агрегирующими серверами auditd, которые обрабатывают события от удалённых систем.
end_of-_event_timeout	2	Неотрицательное количество секунд, после которого событие считается завершённым при анализе потока журнала событий пользовательскими утилитами и библиотечными функциями, такими как aureport (man aureport), ausearch (man ausearch) и т.д. Если в процессе обработки событий время текущего события превышает end_of_event_timeout относительно соседних событий в потоке, то такое событие будет считаться завершённым.
tcp_max_per_addr	1	Числовое значение от 1 до 1024, которое определяет максимально разрешённое количество одновременных подключений с одного IP-адреса. Установка слишком большого значения может привести к DDoS-атаке на сервер auditd. Следует иметь ввиду, что в ядре Linux есть свои собственные лимиты, которые могут ограничить количество подключений даже если настройка сервиса auditd позволяет использовать больше соединений. Значение по умолчанию 1 является достаточным для большинства случаев, если только вы не реализовываете самостоятельно какую-то дополнительную надстройку для пересылки ранее неотправленных сообщений.
tcp_client_ports	Не задано	Определяет с какого исходящего TCP-порта или портов разрешены входящие подключения к сервису auditd. Допустимый диапазон портов: от 1 до 65535. Единичный порт задаётся одним числовым значением, а диапазон — двумя значениями, разделёнными символом «-». Например, чтобы потребовать от клиента использовать привилегированный порт, можно задать значение 1-1023 — это может рассматриваться как дополнительная мера защиты, позволяющая исключить атаки типа «инъекция логов» от имени непривилегированных пользователей. В случае использования этой опции также потребуется установить соответствующее значение директиве local_port в конфигурационном файле /etc/audit/audisp-remote.conf (см. man audisp-remote.conf). В конфигурации по умолчанию никаких ограничений по исходящим портам не применяется.
plugin_dir	/etc/audit/plugins.d	Задаёт каталог, в котором auditd будет осуществлять поиск конфигурационных файлов своих расширений (плагинов).
tcp_client-_max_idle	0	Задаёт время в секундах в течении которого допускается отсутствие каких-либо данных со стороны клиента. Этот параметр используется для закрытия неактивных подключений, если на клиентской системе возникла проблема и она не может самостоятельно завершить подключение. Это глобальная настройка и её значение должно быть выше чем значение heartbeat_timeout (man audisp-remote.conf) на клиентских машинах. Рекомендуется устанавливать значение tcp_client_max_idle в два раза большим чем heart-beat_timeout. Значение по умолчанию 0 отключает эту проверку.
transport	TCP	Если установлено в TCP, то данные между клиентом и сервером auditd будут передаваться в виде открытого текста без шифрования. Если установлено в KRB5, то протокол Kerberos 5 будет использоваться для аутентификации и шифрования.
krb5_key_file	/etc/audit/audit.key	Путь к файлу с ключом для Kerberos-принципала этого клиента. Этот файл должен принадлежать пользователю root и иметь права 0400.
distribute_network	no	Если установлено в yes, то события, поступающие из сети, будут переданы диспетчеру auditd для обработки расширениями (плагинами), что позволит реализовать их дальнейшую пересылку на другой сервер или в систему мониторинга/анализа событий. Если значение no, то события будут только сохраняться в журнал на диске.

продолжение на следующей странице

Таблица 18 – продолжение с предыдущей страницы

Параметр	*Значение по умолчанию*	*Описание*
`overflow_action`	`SYSLOG`	Определяет как служба `auditd` должна реагировать на переполнение внутренней очереди событий. Когда это происходит, это означает, что в очередь на регистрацию поступает больше событий, чем может быть обработано дочерними процессами `auditd`. Эта ошибка также означает, что текущее событие, поступившее на обработку, будет потеряно. Допустимые значения: `IGNORE`, `SYSLOG`, `SUSPEND`, `SINGLE` и `HALT`. Поведение для каждого из значений рассмотрено в описании к директиве `space_left_action`.

Допустимые значения параметров

log_format:

- `RAW` — записи о событиях безопасности будут храниться в том формате, в котором их отправляет ядро операционной системы.
- `ENRICHED` — перед сохранением на диск записи о событиях безопасности будут приведены к более понятному человеку виду путём «разворачивания» информации об идентификаторе пользователя (`uid`), идентификаторе группы (`gid`), системном вызове (`syscall`), архитектуре и адресе сокета. Это упростит анализ событий, созданных на одной системе, в другой системе.

flush:

- `NONE` — не выполнять какие-либо дополнительные действия по синхронизации буфера с диском со стороны службы регистрации событий.
- `INCREMENTAL` — выполнять принудительную синхронизацию буфера на диск с частотой, определяемой параметром `freq`.
- `INCREMENTAL_ASYNC` — поведение похоже на `INCREMENTAL`, но синхронизация буфера выполняется в асинхронном режиме для улучшения производительности.
- `DATA` — незамедлительно синхронизировать данные файла на диск.
- `SYNC` — незамедлительно сохранять данные и метаданные файла на диск.

name_format:

- `NONE` — имя компьютера не будет вставляться в записи журнала безопасности.
- `HOSTNAME` — в журнал будет добавляться имя компьютера, получаемое из системного вызова `gethostname`.
- `FQD` — система аудита получит имя компьютера и преобразует его в полное имя компьютера (FQDN) с помощью DNS-запроса.
- `NUMERIC` — поведение похоже на режим `FQD`, но по имени компьютера будет определяться IP адрес компьютера. Перед использованием этой опции рекомендуется проверить что команды `hostname -i` или `domainname -i` возвращают корректный IP-адрес системы. Использование этой опции не рекомендуется, если для настройки сети используется DHCP так как существует вероятность смены IP-адреса компьютера.

- `USER` — будет использоваться имя, заданное системным администратором в параметре name.

max_log_file_action:

- `IGNORE` — не контролировать максимальный размер файла.
- `SYSLOG` — записать соответствующее предупреждение в системный журнал ОС.
- `SUSPEND` — прекратить записывать журнал аудита на диск, сервис `auditd` при этом продолжит свою работу.
- `ROTATE` — выполнить ротацию файла журнала событий безопасности, при необходимости удалить старые файлы журналов. Будет создан новый файл журнала, а текущий будет переименован — к его имени будет добавлен постфикс `1`, постфикс более старых файлов журналов будет увеличен на единицу. Такого же поведения придерживается утилита `logrotate`.
- `KEEP_LOGS` — поведение аналогично опции `ROTATE`, но старые файлы журналов не будут удаляться, что предотвратит потерю данных из журнала событий безопасности. Когда на диске закончится место, будет выполненное действие, определённое директивой `space_left_action`. Этот вариант рекомендуется использовать совместно с системой резервного копирования.

space_left_action:

- `IGNORE` — ничего не предпринимать.
- `SYSLOG` — записать соответствующее предупреждение в системный журнал.
- `ROTATE` — осуществить ротацию файлов журналов, удалить самые старые из них чтобы освободить место.
- `EMAIL` — отправить соответствующее предупреждение на адрес электронной почты, указанный в директиве `action_mail_acct` и записать предупреждение в системный журнал.
- `EXEC /путь-к-программе` — приостановить запись событий в журнал и выполнить указанную программу, передача параметров программе не поддерживается. Вызванная программа должна освободить место и подать сигнал `SIGUSR2` сервису `auditd` чтобы он возобновил запись событий. Самый простой способ это сделать — выполнить команду `auditctl --signal USR2`.
- `SUSPEND` — прекратить запись данных на диск, при этом сам сервис `auditd` будет активен.
- `SINGLE` — перевести компьютер в однопользовательский режим (также известный как режим восстановления), как если бы системный администратор выполнил команду `telinit 1` или `systemctl isolate runlevel1.target`.
- `HALT` — выключить компьютер. Все действия кроме `ROTATE` будут выполняться только один раз.

disk_error_action:

- `IGNORE` — ничего не предпринимать.
- `SYSLOG` — записать не более пяти последовательных предупреждений в системный журнал.
- `EXEC /путь-к-программе` — приостановить запись событий в журнал и выполнить указанную программу, передача параметров программе не поддерживается. Вызванная программа должна освободить место и подать сигнал `SIGUSR2` сервису `auditd` чтобы он возобновил запись событий. Самый простой способ это сделать — выполнить команду `auditctl --signal USR2`.
- `SUSPEND` — прекратить запись данных на диск, при этом сам сервис `auditd` будет активен.
- `SINGLE` — перевести компьютер в однопользовательский режим (также известный как режим восстановления), как если бы системный администратор выполнил команду `telinit 1` или `systemctl isolate runlevel1.target`.
- `HALT` — выключить компьютер.

Применение изменений

После изменения настроек в конфигурационном файле потребуется применить их, выполнив следующую команду:

```
$ sudo auditctl --signal HUP
```

Служба `auditd` перечитает конфигурационный файл и, если не обнаружит синтаксических ошибок, попробует применить запрошенные изменения. В случае успешного завершения операции в журнале событий безопасности `/var/log/audit/audit.log` появится соответствующее сообщение типа `DAEMON_CONFIG`:

```
type=DAEMON_CONFIG msg=audit(1731317470.852:545): op=reconfigure state=changed \
auid=1000 pid=15069 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 \
res=success AUID="user"
```

В случае возникновения ошибки, в зависимости от её типа, будет выполнено одно из действий, определённых директивами `space_left_action`, `admin_space_left_action`, `disk_full_action` или `disk_error_action` в конфигурационном файле.

Рекомендации по безопасной настройке

В конфигурации, поставляемой по умолчанию в МСВСфера ОС, служба регистрации событий безопасности имеет сбалансированную с точки зрения производительности и безопасности настройку, которая должна подходить под большинство задач.

Ниже приведены некоторые рекомендации по повышению безопасности и отказоустойчивости службы аудита.

- На сервере рекомендуется настроить почтовую систему (MTA, Mail Transfer Agent) для отправки уведомлений от службы аудита системному администратору. В репозиториях МСВСфера ОС доступны следующие почтовые сервера: `postfix`, `sendmail` и `esmtp`. Если ваша система мониторинга использует протокол SNMP (Simple Network Management Protocol), то в качестве альтернативного способа доставки уведомлений можно реализовать соответствующие скрипты/утилиты для отправки предупреждений в систему мониторинга, а через неё, в свою очередь, отправлять предупреждения системному администратору. В состав пакета `net-snmp-utils` входит утилита `snmptrap`, которую можно использовать для отправки сигналов SNMP-trap. Через разработку собственных утилит можно реализовать отправку уведомлений с использованием других протоколов.

- Каталог, в котором хранятся журналы безопасности (см. описание параметра `log_file`), должен находиться на отдельном разделе/точке монтирования. Это позволит избежать ситуации, когда для журналов безопасности не осталось свободного места по причине того, что оно занято файлами других процессов. Также это позволит службе аудита точнее определять и контролировать оставшееся место.

- Параметрам `max_log_file` и `num_logs` должны быть присвоены такие значения, чтобы служба аудита могла использовать всё доступное место на файловой системе. Следует иметь в виду, что чем больше файлов необходимо ротировать, тем больше времени на это потребуется службе аудита прежде чем приступить к записи событий в новый файл журнала. Соответственно, не рекомендуется устанавливать слишком маленький размер файла журнала.

- Параметру `max_log_file_action` рекомендуется присвоить значение `KEEP_LOGS`, чтобы старые файлы журналов безопасности не удалялись.

- Параметру `space_left` рекомендуется установить такое значение, которое оставит системному администратору достаточно времени чтобы среагировать на предупреждение и освободить дисковое пространство. Как правило, в таких случаях требуется выполнить процедуру архивирования файлов журналов.

- Для параметра `space_left_action` рекомендуется установить значение `EMAIL`, чтобы отправить соответствующее предупреждение на электронную почту системному администратору. В качестве альтернативы можно использовать значение `EXEC` и вызывать соответствующую утилиту для отправки уведомления с использованием другого протокола, допустим, `SNMP`.

- Параметру `admin_space_left` рекомендуется установить такое значение, при котором у службы аудита останется достаточно свободного пространства для протоколирования действий системного администратора.

- Значение параметра `admin_space_left_action` рекомендуется установить в `SINGLE`, чтобы перевести систему в однопользовательский режим и позволить администратору освободить место на диске.

- Параметру `disk_full_action` рекомендуется установить значение `HALT` для автоматического выключения системы, если на диске кончилось место — это позволит избежать незапротоколированных действий в системе. Если выключение системы неприемлемо, то установите значение `SINGLE`, чтобы перевести систему в однопользовательский режим.
- Параметр `disk_error_action` должен быть установлен в значение `SYSLOG`, `SINGLE` или `HALT`, в зависимости от политики безопасности вашего предприятия в отношении аппаратных сбоев.
- Если у вас отсутствует система резервного питания и потенциальная утеря нескольких (см. описание директивы `freq` в конфигурационном файле) последних записей о событиях безопасности является для вас критичной, то установите значение директивы `flush` в `SYNC`, что обеспечит постоянную синхронизацию данных и метаданных на диск. Однако, следует отметить, что это приведёт к снижению производительности дисковой подсистемы.

В случае использования сервиса `auditd` для агрегации журналов с других машин по сети рекомендуется:

- Использовать протокол Kerberos для аутентификации и шифрования соединения между компьютерами, за это отвечают директивы `transport`, `krb5_principal` и `krb5_key_file` в конфигурационном файле.
- Разрешить отправку журналов безопасности только с использованием привилегированных портов — директива `tcp_client_ports` в конфигурационном файле. Это поможет избежать атак со стороны непривилегированных пользователей на клиентских машинах.

Управление правилами аудита

Правила фильтрации событий безопасности можно условно разделить по времени их жизни на временные и постоянные. Временные правила в основном используются для отладки, они добавляются в систему вручную, с помощью утилиты `auditctl`, и действуют до перезагрузки или выключения системы. Постоянные правила хранятся в виде файлов в каталоге `/etc/audit/rules.d` и автоматически применяются каждый раз при загрузке системы.

Утилита auditctl

Команда `auditctl` служит для управления базовыми функциями подсистемы аудита, а также позволяет задавать правила, определяющие какие события будут регистрироваться в журнале событий безопасности.

Как и для большинства команд в среде GNU/Linux, для вызова утилиты `auditctl` используется стандартный синтаксис:

```
auditctl [аргументы]
```

Аргументы, которые принимает команда, разбиты на три группы по их назначению.

- Конфигурационные опции — отвечают за настройку параметров ядра, связанных с подсистемой аудита.
- Опции состояния — отображают состояние подсистемы аудита, также существует опция для отправки сообщения в поток событий безопасности.
- Опции для управления правилами фильтрации событий безопасности.

Конфигурационные опции перечислены в таблице.

Таблица 19: Конфигурационные опции `auditctl`

Аргумент	*Описание*
`-b <количество>`	Устанавливает лимит на максимальное количество ожидающих обработки событий безопасности для подсистемы аудита в ядре. Если лимит будет превышен, то ядром будет выставлен флаг сбоя для дальнейшей обработки. Следует иметь в виду, что очередь необработанных сообщений хранится в оперативной памяти, соответственно, объём потребляемой памяти будет пропорционален количеству записей в очереди. Одна запись может занимать около 10 килобайт. По умолчанию размер буфера в ядре равен `64` записям, но правила, поставляемые с МСВСфера ОС, устанавливают его размер в `8192` записей.
`--backlog_wait_time <время_ожидания>`	Задаёт максимальное время ожидания пока размер полностью заполненной очереди событий, ожидающих обработки, не уменьшится. В случае превышения этого лимита текущее событие, ожидающее обработки, будет утеряно (удалено). При этом в журнал событий будет записана соответствующая ошибка. По умолчанию таймаут в ядре равен `60×HZ`.
`--reset_backlog-_wait_time_actual`	Сбрасывает счётчик фактического времени ожидания уменьшения очереди событий, отображаемый статус-командой `auditctl -s`.
`-c`	Продолжать загружать правила несмотря на возникающие ошибки. Сохраняет результаты загрузки всех правил и, если хотя бы одно правило не загрузилось, то код возврата будет не нулевой.
`-D`	Удалить все правила и точки наблюдения. Может быть скомбинирован с аргументом `-k`.
`-e [0..2]`	Управляет состоянием службы аудита. Допустимые значения параметра перечислены после таблицы.
`-f [0..2]`	Устанавливает способ обработки возникающих сбоев. Флаг сбоя устанавливается при ошибках передачи данных из пространства ядра в службу `auditd`, работающую в пользовательском пространстве, при превышении максимального времени обработки очереди событий, нехватке памяти и т.п. Допустимые значения параметра перечислены после таблицы.
`-h`	Выдать справочную информацию об аргументах командой строки и завершить работу.
`-i`	Если используется самостоятельно, то включает режим игнорирования ошибок при чтении правил из файла — в таком случае `auditctl` всегда будет возвращать успешный код возврата. Если же используется в комбинации с аргументом `-s`, то, по возможности, переводит числовые идентификаторы в понятные пользователю слова.
`--loginuidimmutable`	Делает UID учётных записей неизменяемым сразу после его установки, что предотвращает возможность выдавать себя за других пользователей. Для изменения UID требуется полномочие `CAP_AUDIT_CONTROL`, поэтому непривилегированный пользователь не может его изменить. Использование этого параметра может вызвать проблемы при использовании контейнеризации.
`-t`	«Обрезать» неиспользуемые ветви поддеревьев каталогов после монтирования.
`-q <точка_монтирования, поддерево>`	При наличии точки наблюдения за каталогом и объединении или перемещении точки монтирования другого поддерева в наблюдаемое, указывает ядру сделать монтируемое поддерево эквивалентным наблюдаемому каталогу. Если поддерево уже смонтировано во время создания точки наблюдения, то оно автоматически помечается как наблюдаемое. Обратите внимание: значения разделяются запятой, её отсутствие вызовет ошибку.
`-r <частота>`	Устанавливает лимит сообщений аудита в секунду. Если значение больше нуля и было превышено, то ядром будет выставлен флаг сбоя для обработки. Значение по умолчанию — `0`, что означает отсутствие ограничений.
`--reset-lost`	Сбрасывает счётчик потерянных записей, отображаемых статус-командой `auditctl -s`.
`-R <путь_к_файлу>`	Считать и выполнить правила из указанного файла. Правила будут применяться построчно, в том порядке, в котором они определены в файле. Файл должен принадлежать пользователю `root` и быть недоступным для чтения и записи другими пользователями, в противном случае запрос на его обработку будет отклонён. Строки, имеющие в начале символ `#` считаются комментариями. Каждая строка будет обрабатываться как набор аргументов для команды `auditctl`. Поскольку файл обрабатывается именно командой `auditctl`, а не командной оболочкой `bash`, не экранируйте специальные символы `shell`. Примеры файлов с правилами приведены дальше в этом разделе.
`--signal <сигнал>`	Отправляет сигнал (`man 7 signal`) службе аудита, для этого потребуются соответствующие привилегии. Поддерживаемые сигналы перечислены после таблицы.

Допустимые значения параметров

-f [0..2]:

- `0` — «тихий» режим, ничего не предпринимать.
- `1` — записать сообщение об ошибке в журнал сообщений ядра с помощью функции `printk()`.

- 2 — отрапортовать о критичной ошибке ядра и перевести его в состояние «kernel panic» заблокировав тем самым дальнейшую работу системы. Значение по умолчанию — 1, но для защищённых окружений рекомендуется использовать 2.

-e [0..2]:

- 0 — временно отключить службу аудита.
- 1 — включить службу аудита.
- 2 — включить службу аудита и заблокировать последующие изменения её конфигурации. Блокировка конфигурации должна быть последней командой в цепочке правил для службы аудита, после её применения любая попытка изменить конфигурацию службы аудита будет запротоколирована и отклонена. Изменение правил вновь станет возможным только после перезагрузки компьютера.

Поддерживаемые сигналы для –signal <сигнал>:

- TERM (stop) — завершает работу службы аудита.
- HUP (reload) — при получении данного сигнала служба auditd перечитает конфигурационный файл и, если не обнаружит синтаксических ошибок, попробует применить запрошенные изменения. В случае успешного завершения операции в журнале событий безопасности /var/log/audit/audit.log появится соответствующее сообщение типа DAEMON_CONFIG. В случае возникновения ошибки, в зависимости от её типа, будет выполнено одно из действий, определённых директивами space_left_action, admin_space_left_action, disk_full_action или disk_error_action в конфигурационном файле.
- USR1 (rotate) — указывает службе аудита на необходимость прекратить запись в текущий файл журнала, создать новый файл и продолжить вести журнал в нём. Этот сигнал может быть полезным для организации процесса резервного копирования.
- USR2 (resume) — указывает службе аудита на необходимость продолжить ведение журнала событий безопасности. Обычно это требуется после приостановки ведения журнала или переполнения внутренней очереди подсистемы аудита.

Опции состояния перечислены в таблице.

Таблица 20: Опции состояния auditctl

Аргумент	*Описание*
-l	Вывести список всех правил по одному на строку. В связке с этой опцией можно использовать ещё два параметра: -k <ключ> — вывести только правила, соответствующие заданному ключу. -i — интерпретировать значения полей от a0 до a3 для корректного определения аргументов системных вызовов.
-m <текст>	Отправить в подсистему аудита сообщение из пользовательского пространства. Это можно сделать только от имени суперпользователя root или если у вашего пользователя есть полномочия CAP_AUDIT_WRITE. Отправленное сообщение будет иметь тип USER.

продолжение на следующей странице

Таблица 20 – продолжение с предыдущей страницы

Аргумент	*Описание*
-s	Показать отчёт о состоянии подсистемы аудита в ядре. В нём будут указаны значения, которые устанавливаются с помощью опций -e, -f, -r и -b команды auditctl. Значение pid — это идентификатор процесса службы аудита, если оно равно 0, то служба аудита не запущена. Значение lost отображает количество записей, которые были удалены из-за переполнения очереди событий, ожидающих обработки. Значение backlog отображает количество записей, которые в данный момент находятся в очереди на обработку. Также к опции -s можно добавить параметр -i чтобы получить интерпретированное значение некоторых числовых полей.
-v	Вывести версию утилиты auditctl.

Опции для управления правилами фильтрации перечислены в таблице.

Таблица 21: Опции для управления правилами фильтрации auditctl

Аргумент	*Описание*
-a <список,действие | действие, список>	Добавить правило с указанным действием в конец списка. Значения должны быть разделены запятой, при этом их порядок не играет роли. Допустимые имена списков и действия для правил перечислены после таблицы.
-A <список,действие>	Добавить правило с указанным действием в начало списка. Информацию по доступным действиям и спискам смотрите выше в описании к директиве -a.
-C <поле=поле | поле!=поле>	Создать условие сравнения двух полей для правила. Используется синтаксис первое_поле оператор второе_поле, поддерживаются операторы: = (равно) и != (не равно). В одном правиле может использоваться несколько операций сравнения, каждая должна начинаться с аргумента -C. Чтобы правило сработало для события аудита, все его условия, заданные директивами -C и -F должны быть выполнены. Сравнение можно выполнять для следующих полей: группа uid: auid, uid, euid, suid, fsuid и obj_uid. группа gid: gid, egid, sgid, fsgid и obj_gid. Сравнивать между собой можно только поля из одной группы. Поля obj_uid и obj_gid заполняются данными из объекта, для которого возникло событие — файла, каталога и т.п..
-d <список,действие>	Удалить правило с указанным действием из списка. Правило будет удалено только если полностью совпали все поля условия и название системного вызова.
-F <поле=значение | поле!=значение | поле<значение | поле>значение | поле<=значение | поле>=значение | поле&значение | поле&=значение>	Создать условие сравнения для правила. Используется синтаксис поле оператор значение, поддерживаются операторы: = (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно), & (битовая маска) и &= (битовая проверка). В одном правиле может быть до 64 условий, каждое должно начинаться с аргумента -F. Чтобы правило сработало для события аудита, все его условия, заданные директивами -F и -C должны быть выполнены. Для полей, содержащих идентификатор пользователя можно также использовать имя пользователя — программа самостоятельно преобразует имя в идентификатор. То же самое выполняется и для полей с идентификатором группы. Допустимые имена полей перечислены после таблицы.
-W <путь>	Удалить точку наблюдения за объектом файловой системы по указанному пути. Требуется полное соответствие правилу, смотрите описание к опции -d.
-k <ключ>	Устанавливает ключ фильтрации для данного правила. Ключом может быть произвольная текстовая строка длиной не больше 31 байта. Ключ позволяет однозначно идентифицировать записи журнала безопасности, созданные с помощью правила. Обычно используется когда у вас есть несколько правил, которые в совокупности удовлетворяют требованию безопасности. По ключу можно отфильтровать все соответствующие записи с помощью утилиты ausearch, не зависимо от того, какое правило сработало. Ключ также можно использовать для удаления определённых правил с помощью аргумента -D или для получения списка соответствующих правил с помощью аргумента -l. К одному правилу может быть привязано несколько ключей если вы хотите выполнять поиск по событиям несколькими способами или если вы используете собственное расширение auditd для анализа записей в журнале.
-p <r|w|x|a>	Устанавливает фильтр прав доступа к объекту файловой системы: r — чтение (read), w — запись (write), x — исполнение (execute), a — изменение атрибутов (attribute change). Не следует путать эти права с обычными правами доступа к файлу, скорее они определяют системные вызовы, которые выполняют данные действия. Обратите внимание, системные вызовы read и write не включены в этот набор, поскольку они перегрузили бы журнал аудита сообщениями.
-S <имя_или_номер_системного_вызова | all>	Название или номер системного вызова, который необходимо отслеживать. Также можно использовать ключевое слово all, которое включает обработку всех системных вызовов. В случае если установлен фильтр по другим полям, а системный вызов не указан, то правило будет применяться ко всем системным вызовам. Используя несколько опций -S можно указать несколько системных вызовов в одном правиле — это повышает производительность, поскольку потребуется вычислять меньшее количество правил. В качестве альтернативы, вы можете указать несколько системных вызовов через запятую. Если вы работаете с системой, которая поддерживает несколько архитектур, например x86_64, то вы должны знать, что auditctl просто берёт название системного вызова, находит номер системного вызова в таблице вызовов для «родной архитектуры» (в данном случае, для b64) и отправляет это правило ядру. Соответственно, если в правиле не определено поле arch, то правило будет применено и к 64-битным системным вызовам, и к 32-битным. Это может привести к нежелательным последствиям, поскольку системные вызова для 32-битных и 64-битных систем могут иметь разные номера. Соответственно, в таком случае рекомендуется создавать два отдельных правила для b32 и b64-архитектур.

продолжение на следующей странице

Таблица 21 – продолжение с предыдущей страницы

Аргумент	*Описание*
`-w <путь>`	Добавить точку наблюдения за объектом файловой системы по указанному пути. Вы не можете создать точку наблюдения за корневым каталогом (`/`) поскольку это запрещено ядром. Групповые символы (wildcards) запрещены — при попытке их использования будет выдано соответствующее предупреждение. Внутри точки наблюдения реализованы методом слежения за номерами индексных дескрипторов (`inodes`). Если вы устанавливаете точку наблюдения за файлом, то это равносильно использованию опции `-F path=значение` в правиле. Если же вы устанавливаете точку наблюдения за каталогом, то это равносильно использованию опции `-F dir=значение` в правиле. Форма записи правил через опцию `-w` предназначена для обратной совместимости, запись через `-F` является более выразительной. В отличии от большинства правил аудита системных вызовов, точки наблюдения за файловой системой не оказывают существенного влияния на производительность в зависимости от количества правил, отправленных ядру. Единственными допустимыми параметрами при использовании опции `-w` являются `-p` и `-k`. Если вам требуется выполнить что-то необычное, например, провести аудит доступа определённого пользователя к файлу, то воспользуйтесь опцией `-F` с аргументами `path` или `dir`.

Допустимые имена списков аргумента -a <список,действие | действие,список>:

- `task` — добавить правило к списку, отвечающему за процессы. Этот список правил используется только во время создания процесса — когда родительский процесс вызывает функции `fork()` или `clone()`. При использовании этого списка вы должны использовать только те поля, которые известны на момент создания процесса: `uid`, `gid` и т.д..
- `exit` — добавить правило к списку, отвечающему за точки выхода из системных вызовов. Этот список используется чтобы определить, следует ли создавать событие аудита при завершении системного вызова.
- `user` — добавить правило в список, отвечающий за фильтрацию пользовательских сообщений. Этот список используется ядром для фильтрации сообщений, исходящих из пользовательского пространства, перед передачей их службе аудита. При использовании этого списка поддерживаются только следующие поля: `uid`, `auid`, `gid`, `pid`, `subj_user`, `subj_role`, `subj_type`, `subj_sen`, `subj_clr`, `msgtype` и `exe`. Все остальные поля будут считаться не соответствующими условию. Следует понимать, что любое событие, исходящее из пространства пользователя, у которого есть полномочие `CAP_AUDIT_WRITE`, будет записано в журнал событий безопасности. Соответственно, в большинстве случаев этот фильтр будет использоваться с правилами, действие для которых — `never`, поскольку для записи события ничего не нужно делать.
- `exclude` — добавить правило, исключающее определённый тип события. Например, таким образом можно отключить протоколирование событий от SELinux AVC. События можно исключать по следующим полям: `pid`, `uid`, `gid`, `auid`, `msgtype`, `subj_user`, `subj_role`, `subj_type`, `subj_sen`, `subj_clr` и `exe`. Значение параметра `действие` игнорируется и всегда используется значение `never`.
- `filesystem` — добавить правило, которое будет применяться ко всем файловым системам заданного в поле `fstype` типа. Обычно этот фильтр используется для исключения всех событий, связанных со специальными файловыми системами, например `debugfs` или `tracefs`.

- io_uring — добавить правило к фильтру системных вызовов подсистемы ядра io_uring. Правила для этого фильтра должны указывать системный вызов с помощью аргумента -S <системный_вызов>, описанного ниже. Вы также можете использовать аргумент -k <ключ> для правила чтобы сгруппировать его с другими правилами, отслеживающими тот же самый системный вызов.

Допустимые действия для правил аргумента -a <список,действие | действие,список>:

- never — не генерировать записи аудита для подходящих под правило событий. В общем случае рекомендуется размещать такие правила в начале списка, так как срабатывает первое подходящее правило.
- always — создавать запись аудита, всегда наполнять её данными в точке входа в системный вызов и выдавать на обработку в момент выхода из системного вызова.

Допустимые имена полей аргумента -F <поле=значение | поле!=значение | поле<значение | поле>значение | поле<=значение | поле>=значение | поле&значение | поле&=значение>:

- a0, a1, a2, a3 — первые четыре аргумента, переданные системному вызову. Строковые аргументы не поддерживаются поскольку ядро получает указатель на строку вместо самой строки. Соответственно, при использовании этих полей, вам следует использовать только числовые значения. Обычно это используется на платформах, мультиплексирующих сокеты или операции IPC.
- arch — архитектура процессора, на котором выполняется системный вызов. Узнать архитектуру системы можно с помощью команды uname -m. Если вы не знаете архитектуру своего компьютера, но хотите использовать 32-разрядные системные вызовы и ваш компьютер поддерживает 32 бита, вы также можете использовать b32 вместо архитектуры. Таким же образом можно использовать b64 для 64-разрядных системных вызовов. Таким образом можно создавать в некотором смысле независимые от архитектуры правила, поскольку тип семейства будет определяться автоматически. Однако, следует помнить, что системные вызовы могут быть специфичными для определённой архитектуры и то, что доступно для x86_64, может быть недоступно на aarch64. Архитектура должна указываться перед аргументом -S чтобы утилита auditctl могла определить в какой внутренней таблице искать номера системных вызовов.
- auid — исходный идентификатор пользователя, использованный для входа в систему. auid — это сокращение от audit uid, также его иногда называют loginuid. В качестве значения может использовать как идентификатор пользователя, так и его имя.
- devmajor — старший номер устройства (device major number).
- devminor — младший номер устройства (device minor number).

- `dir` — полный путь к каталогу для наблюдения. Это приведёт к рекурсивному просмотру этого каталога и всего его поддерева. Это поле можно использовать только в списке `exit`. Также смотрите описание опции `-w`.
- `egid` — действующий идентификатор группы. Может использоваться как числовой идентификатор, так и название.
- `euid` — действующий идентификатор пользователя. Может использоваться как числовой идентификатор, так и имя.
- `exe` — абсолютный путь к приложению, для которого будет применяться это правило. Поддерживаются только операторы `=` и `!=`. Это поле можно проверять только один раз для каждого правила.
- `exit` — код возврата из системного вызова. Если в качестве кода используется ошибка `errno` (`man 3 errno`), то можно использовать её текстовое представление.
- `fsgid` — идентификатор группы, применяемый к файловой системе.
- `fstype` — тип файловой системы, используется только в списке правил `filesystem`. Допустимые значения: `debugfs` и `tracefs`.
- `fsuid` — идентификатор пользователя, применяемый к файловой системе. Можно использовать как числовой идентификатор, так и имя пользователя.
- `filetype` — тип целевого файла. Допустимые значения: `file`, `dir`, `socket`, `link`, `character`, `block` или `fifo`.
- `gid` — идентификатор группы. Можно использовать как числовой идентификатор, так и название группы.
- `inode` — номер индексного дескриптора (`inode`).
- `key` — альтернативный способ установки ключа для фильтрации правил. Смотрите описание опции `-k` ниже.
- `msgtype` — тип сообщения, к которому должно применяться правило. Может использоваться только в списках `exclude` и `user`.
- `obj_uid` — идентификатор пользователя, связанный с объектом (файлом или каталогом).
- `obj_gid` — идентификатор группы, связанный с объектом (файлом или каталогом).
- `obj_user` — имя SELinux пользователя, владеющего ресурсом.
- `obj_role` — SELinux роль ресурса.
- `obj_type` — SELinux тип ресурса.
- `obj_lev_low` — нижний уровень ресурса в контексте SELinux.
- `obj_lev_high` — верхний уровень ресурса в контексте SELinux.

- `path` — полный путь к файлу для точки наблюдения, используется только в списке правил `exit`.
- `perm` — фильтр прав доступа для файловых операций. Подробное описание доступно в справке по опции `-p`. Этот фильтр используется только в списке правил `exit`. Его также можно использовать без указания системного вызова — ядро само подберёт системные вызовы, которые удовлетворяют запрашиваемым разрешениям.
- `pers` — персональный номер операционной системы.
- `pid` — идентификатор процесса.
- `ppid` — идентификатор родительского процесса.
- `saddr_fam` — номер семейства протоколов, который указан в файле `/usr/include/bits/socket.h`. Например, IPv4 будет иметь номер 2, а IPv6 — 10.
- `sessionid` — идентификатор сеанса пользователя.
- `subj_user` — имя пользователя-владельца процесса в контексте SELinux.
- `subj_role` — SELinux роль процесса.
- `subj_type` — SELinux тип процесса.
- `subj_sen` — SELinux чувствительность процесса (Linux Sensitivity).
- `subj_clr` — SELinux допуск процесса (SELinux Clearance).
- `sgid` — установленный идентификатор группы (см. `man getresgid`).
- `success` — если код возврата системного вызова больше либо равен нулю, то данное поле будет иметь значение `true/yes`, иначе — `false/no`. При создании правила используйте 1 для `true/yes` и 0 для `false/no`.
- `suid` — установленный идентификатор пользователя (см. `man getresuid`).
- `uid` — идентификатор пользователя, можно использовать как числовой идентификатор, так и имя пользователя.

Создание правил аудита

В этом разделе приведены примеры решения некоторых типовых задач, связанных с регистрацией событий безопасности. В большинстве примеров используются временные правила для подсистемы аудита, процедура создания постоянных правил описана в следующем разделе.

Контроль загрузки и выгрузки модулей ядра

Добавление и/или удаление модулей ядра может быть использовано для изменения поведения ядра и внедрения вредоносного кода в пространство ядра. Приведённый ниже набор правил позволит отслеживать такие операции:

```
# регистрировать любой запуск команды /usr/bin/kmod. В МСВСфера ОС 9 команды
# /usr/sbin/insmod, /usr/sbin/rmmod и /usr/sbin/modprobe являются символическими
# ссылками на /usr/bin/kmod.
$ sudo auditctl -a always,exit -F path=/usr/bin/kmod -F perm=x \
  -F key=kernel_modules

# предыдущее правило так же можно переписать с использованием опции `-w`,
# с точки зрения подсистемы аудита оба правила эквивалентны:
# sudo auditctl -w /usr/bin/kmod -p x -k kernel_modules

# отслеживать системные вызовы, выполняющие загрузку и выгрузку модулей ядра
# на 64-битной архитектуре
$ sudo auditctl -a always,exit -F arch=b64 \
  -S init_module,finit_module,delete_module -F key=kernel_modules
```

Для всех подпадающих под правило записей в журнале событий безопасности будет устанавливаться ключ `kernel_modules`. Соответственно, для их просмотра можно использовать следующую команду:

```
$ sudo ausearch -k kernel_modules
```

Отслеживание изменений в конфигурации sudo

`Sudo` — это утилита, позволяющая пользователю выполнять программы с привилегиями другой учётной записи, в том числе и от имени суперпользователя. Настройка `sudo` выполняется либо через редактирование основного конфигурационного файла `/etc/sudoers`, либо через создание дополнительных конфигурационных файлов в каталоге `/etc/sudoers.d`.

Следующий набор правил позволит отслеживать изменения в конфигурации `sudo`:

```
# отслеживать любые изменения файла /etc/sudoers
$ sudo auditctl -a always,exit -F path=/etc/sudoers -F perm=wa \
  -F key=sudo_config

# отслеживать любые изменения в каталоге /etc/sudoers.d
$ sudo auditctl -a always,exit -F dir=/etc/sudoers.d -F perm=wa \
  -F key=sudo_config
```

Эти правила также можно записать с использованием опции `-w`:

```
# отслеживать любые изменения файла /etc/sudoers
$ sudo auditctl -w /etc/sudoers -p wa -k sudo_config

# отслеживать любые изменения в каталоге /etc/sudoers.d
$ sudo auditctl -w /etc/sudoers.d -p wa -k sudo_config
```

Для просмотра записей можно будет использовать использовать следующую команду:

```
$ sudo ausearch -k sudo_config
```

Отслеживание установки или обновления RPM-пакетов

Для отслеживания установки или обновления RPM-пакетов подсистемой аудита установите из репозиториев МСВСфера ОС пакет `rpm-plugin-audit`, который содержит необходимое расширение для пакетного менеджера RPM:

```
$ sudo dnf install -y rpm-plugin-audit
```

Больше никаких действий по настройке не требуется — при каждой установке или обновлении RPM-пакета в журнал событий безопасности будет добавляться соответствующая запись с типом `SOFTWARE_UPDATE`.

Пример сообщения об обновлении пакета `osinfo-db`:

```
type=SOFTWARE_UPDATE msg=audit(1731661320.537:3549): \
pid=22903 uid=0 auid=1000 ses=4 subj=unconfined_u:unconfined_r: \
unconfined_t:s0-s0:c0.c1023 msg='op=update sw="osinfo-db-20231215-1.el9.inferit.noarch" \
sw_type=rpm key_enforce=0 gpg_res=1 root_dir="/" comm="dnf" exe="/usr/bin/python3.9" \
hostname=msvsphere-94-arm.msvsphere.test addr=? terminal=pts/2 res=success'
```

Отслеживание установки модулей для Lua, NodeJS, Perl, Python, Ruby

Многие современные языки программирования имеют собственный пакетный менеджер для установки дополнительных библиотек или утилит из централизованных репозиториев. В набор правил подсистемы аудита МСВСфера ОС входят правила для отслеживания запуска следующих пакетных менеджеров:

- `/usr/bin/pip` — установщик Python-модулей
- `/usr/bin/npm` — установщик NodeJS-модулей
- `/usr/bin/cpan` — установщик Perl-модулей
- `/usr/bin/gem` — установщик Ruby-модулей
- `/usr/bin/luarocks` — установщик Lua-модулей
- `/usr/bin/dnf-3` — пакетный менеджер DNF
- `/usr/bin/yum` — пакетный менеджер Yum (в настоящее время является ссылкой на DNF чтобы обеспечить совместимость)

Данные правила находятся в файле `/usr/share/audit/sample-rules/44-installers.rules` и не включены по умолчанию. Для их активации выполните следующие действия.

Примечание

В МСВСфера 10 файлы `.rules` лежат по пути `/usr/share/audit-rules/`.

1. Скопируйте файл с правилами в каталог постоянных правил подсистемы аудита `/etc/audit/rules.d`:

```
$ sudo cp /usr/share/audit/sample-rules/44-installers.rules /etc/audit/rules.d/
```

2. Загрузите обновлённый набор правил в подсистему аудита:

```
$ sudo augenrules --load
```

3. Убедитесь, что правила были загружены, с помощью команды:

```
$ sudo auditctl -l | grep software-installer
-p x-w /usr/bin/dnf-3 -k software-installer
-p x-w /usr/bin/yum -k software-installer
-p x-w /usr/bin/pip -k software-installer
-p x-w /usr/bin/npm -k software-installer
-p x-w /usr/bin/cpan -k software-installer
-p x-w /usr/bin/gem -k software-installer
-p x-w /usr/bin/luarocks -k software-installer
```

Теперь при запуске любого из отслеживаемых пакетных менеджеров в журнал подсистемы аудита будет добавлена соответствующая запись. Отфильтровать такие события можно по ключу `software-installer`. Ниже приведён пример события, которое было запротоколировано при выполнении команды `pip install markdown2`:

```
$ ausearch -k software-installer
----
time->Fri Nov 15 13:40:38 2024
type=PROCTITLE msg=audit(1731667238.489:3725):␣
↪proctitle=2F7573722F62696E2F707974686F6E33002F62696E2F70\
697000696E7374616C6C006D61726B646F776E32
type=PATH msg=audit(1731667238.489:3725): item=2 name="/lib64/ld-linux-x86-64.so.2" \
inode=1980964 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:ld_so_t:s0 \
nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1731667238.489:3725): item=1 name="/usr/bin/python3" inode=1993087 \
dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 \
nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1731667238.489:3725): item=0 name="/bin/pip" inode=1966587 \
dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:bin_t:s0 \
nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1731667238.489:3725): cwd="/home/user"
type=EXECVE msg=audit(1731667238.489:3725): argc=4 a0="/usr/bin/python3" \
a1="/bin/pip" a2="install" a3="markdown2"
type=SYSCALL msg=audit(1731667238.489:3725): arch=c000003e syscall=59 \
success=yes exit=0 a0=560b1acfe030 a1=560b1aceae30 a2=560b1abd46e0 a3=8 \
items=3 ppid=14961 pid=23067 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 \
sgid=0 fsgid=0 tty=pts2 ses=4 comm="pip" exe="/usr/bin/python3.9" \
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="software-installer"
```

Более подробно работа с постоянными правилами подсистемы аудита и утилитой `ausearch` описана в следующих разделах.

Рекомендации по оптимизации правил

Правила, анализирующие системные вызовы, проверяются для каждого системного вызова каждой выполняемой программы. Соответственно, если у вас десять правил для системных вызовов, то каждая программа во время выполнения любого системного вызова будет приостановлена пока подсистема аудита применяет каждое из этих правил. По возможности старайтесь комбинировать как можно больше системных вызовов в одном правиле, если фильтры, действия и ключ для этих системных вызовов совпадают.

В качестве примера рассмотрим следующие три правила, которые отслеживают системные вызовы, выполняющие загрузку и выгрузку модулей ядра:

```
$ sudo auditctl -a always,exit -F arch=b64 -S init_module -F key=kernel_modules
$ sudo auditctl -a always,exit -F arch=b64 -S finit_module -F key=kernel_modules
$ sudo auditctl -a always,exit -F arch=b64 -S delete_module -F key=kernel_modules
```

Поскольку все остальные поля правила совпадают, их легко можно объединить в одно правило, которое уже приводилось в примере ранее:

```
$ sudo auditctl -a always,exit -F arch=b64 \
  -S init_module,finit_module,delete_module -F key=kernel_modules
```

Также можно указывать каждый системный вызов с помощью отдельного аргумента `-S` — с точки зрения подсистемы аудита оба варианта эквивалентны:

```
$ sudo auditctl -a always,exit -F arch=b64 \
  -S init_module -S finit_module -S delete_module -F key=kernel_modules
```

Также постарайтесь использовать точки наблюдения за файловой системой там, где это возможно. Это значительно повышает производительность, поскольку правила будут применяться только для операций с указанными файлами и каталогами, а не ко всем системным вызовам.

В качестве примера предположим, что вы бы хотели фиксировать все неудачные операции открытия и усечения (truncate) файлов. Тогда вы могли бы написать следующее правило:

```
$ auditctl -a always,exit -F arch=b64 -S openat -S truncate -F success=0
```

Такое правило применялось бы ко всем системным вызовам всех программ, независимо от того, какие файлы они изменяют. Будут отслеживаться и изменения временных файлов в каталоге `/tmp`, и изменения файлов в домашних каталогах пользователя и любые другие операции.

Но, вероятнее всего, вместо любых операций вы хотели бы отслеживать изменения конкретных файлов, допустим, общесистемных файлов конфигурации в каталоге `/etc`. Тогда правило будет иметь следующий вид:

```
$ auditctl -a always,exit -F dir=/etc -F arch=b64 -S openat,truncate -F success=0
```

И будет применяться только к файлам внутри каталога `/etc` или в его подкаталогах, что значительно сократит количество проверяемых системных вызовов.

Создание постоянных правил

Постоянные правила подсистемы аудита хранятся в виде файлов в каталоге `/etc/audit/rules.d/` и автоматически применяются во время запуска службы `auditd`.

Каждый файл с правилами должен иметь расширение `.rules`, например, `50-mail-server.rules`. Файлы с правилами обрабатываются и загружаются последовательно, в порядке естественной сортировки («natural sort order»). Общепринятой, хотя и не обязательной, является следующая схема именования файлов: `приоритет-описание.rules` где `приоритет` — целое число, задающее порядок загрузки файла, а `описание` — краткое описание назначения загружаемых правил.

При разработке правил для МСВСфера ОС предлагается придерживаться следующей схемы:

- `10` — правила для настройки ядра и подсистемы аудита;
- `20` — правила, переопределяющие настройки ядра и подсистемы аудита, поставляемые с операционной системой;
- `30` — основные правила;
- `40` — дополнительные/опциональные правила;
- `50` — правила, специфичные для группы серверов;
- `70` — правила, специфичные для локальной системы;
- `90` — правило, блокирующее дальнейшее изменение списка правил.

Файл с правилами подсистемы аудита является обычным текстовым файлом, в котором используется следующий формат:

- пустые строки и весь текст после символа `#` игнорируются;
- каждая непустая строка считается правилом и оформляется в виде списка аргументов для команды `auditctl` (см. раздел audit-rules-creation), при этом сама команда `auditctl` не указывается — она будет автоматически вызвана системой с заданными аргументами при обработке файла с правилами. На одной строке должно объявляться только одно правило;
- файл должен оканчиваться пустой строкой.

Пример файла с правилами, отслеживающими изменения в конфигурационных файлах системы виртуализации на базе гипервизора `libvirt`:

```
-w /etc/libvirt/libvirt-admin.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/libvirt.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/libvirtd.conf -p wa -k libvirt-config-changes
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
-w /etc/libvirt/qemu.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/qemu-lockd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtinterfaced.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtlockd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtlogd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtnetworkd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtnodedevd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtnwfilterd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtproxyd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtqemud.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtsecretd.conf -p wa -k libvirt-config-changes
-w /etc/libvirt/virtstoraged.conf -p wa -k libvirt-config-changes
-w /usr/share/polkit-1/actions/org.libvirt.api.policy -p wa -k libvirt-polkit-changes
-w /usr/share/polkit-1/actions/org.libvirt.unix.policy -p wa -k libvirt-polkit-changes
-w /usr/share/polkit-1/rules.d/50-libvirt.rules -p wa -k libvirt-polkit-changes
```

С точки зрения безопасности рекомендуется назначать владельцем файлов с правилами пользователя `root`, группу `root` и разрешать доступ на чтение и запись только пользователю `root`:

```
$ sudo chown root:root /etc/audit/rules.d/50-mail-server.rules
$ sudo chmod 600 /etc/audit/rules.d/50-mail-server.rules
```

Как уже упоминалось ранее, правила из файлов загружаются автоматически при запуске службы `auditd`. Однако, в случае необходимости, можно загрузить новые правила с помощью следующей команды:

```
$ sudo augenrules --load
```

Работа с журналом событий безопасности

Формат файла журнала событий безопасности

В конфигурации по умолчанию подсистема аудита хранит текущий журнал событий безопасности в файле `/var/log/audit/audit.log`. Если ротация файлов журналов включена, то предыдущие файлы журнала будут находиться в том же каталоге.

Рассмотрим формат записей журнала на примере, для этого добавим в подсистему аудита следующее правило, которое будет регистрировать операции чтения файла `/etc/fstab`:

```
$ sudo auditctl -a always,exit -F path=/etc/fstab -F perm=r -F key=fstab_read
```

Теперь прочитаем файл какой-нибудь командой, допустим `cat`:

```
$ cat /etc/fstab
```

В журнале `/var/log/audit/audit.log` появятся следующие записи о событии:

```
type=SYSCALL msg=audit(1731576783.677:1528): arch=c000003e syscall=257 \
success=yes exit=3 a0=ffffff9c a1=7ffd91e5b610 a2=0 a3=0 items=1 ppid=19085 \
pid=20721 auid=1667 uid=1667 gid=1667 euid=1667 suid=1667 fsuid=1667 \
egid=1667 sgid=1667 fsgid=1667 tty=pts6 ses=7 comm="cat" exe="/usr/bin/cat" \
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="fstab_read"

type=CWD msg=audit(1731576783.677:1528): cwd="/home/user"

type=PATH msg=audit(1731576783.677:1528): item=0 name="/etc/fstab" \
inode=262528 dev=fd:00 mode=0100644 ouid=0 ogid=0 rdev=00:00 \
obj=unconfined_u:object_r:etc_t:s0 nametype=NORMAL cap_fp=0 cap_fi=0 \
cap_fe=0 cap_fver=0 cap_frootid=0 OUID="root" OGID="root"

type=PROCTITLE msg=audit(1731576783.677:1528): proctitle=636174002F6574632F6673746162
```

Все четыре записи имеют одинаковое значение поля `msg`, что определяет их принадлежность к одному событию. Записи журнала аудита всегда начинаются с поля `type`, каждая запись состоит из нескольких пар `поле=значение`, разделённых пробелом, в некоторых случаях может использоваться запятая.

Теперь рассмотрим каждую из записей подробнее.

Для первой записи установлены следующие поля.

- `type=SYSCALL` — поле `type` указывает на тип записи, в данном случае значение `SYSCALL` означает, что эта запись вызвана системным вызовом ядра.
- `msg=audit(1731576783.677:1528)` — поле `msg` содержит следующие данные.
  - Временную отметку и уникальный идентификатор события в формате `audit(временная_отметка:ID)`. Несколько записей в журнале аудита могут иметь одинаковую отметку и идентификатор если они были сгенерированы в рамках обработки одного события безопасности. Для временной отметки используется стандартный для Unix-подобных систем формат — количество секунд, прошедших с 00:00:00 по UTC 1 января 1970 года. Существует множество способов преобразования такой временной отметки в понятный человеку формат, самый простой из них — утилита `date`:

    ```
    $ date -d @1731576783.677
    Чт 14 ноя 2024 12:33:03 MSK
    ```

  - Различные специфичные для события пары `поле=значение`, полученные из ядра или из пользовательского пространства. Формально, все данные, которые находятся после `msg=audit(временная_отметка:ID): ...` являются значением поля `msg`, так что не удивляйтесь если в некоторых случаях вы даже увидите два поля `msg` в одной записи — такие записи встречаются, например, в событиях от гипервизора `libvirt`.
- `arch=c000003e` — информация об архитектуре системы, закодированная в шестнадцатеричной системе. При поиске записей с помощью утилиты ausearch используйте аргумент `-i / --interpret` чтобы автоматически преобразовывать закодированные значения в понятный человеку формат. Значение `c000003e` будет преобразовано в `x86_64`.
- `syscall=257` — тип (номер) системного вызова, который был отправлен ядру. Для 64-битной архитектуры понятное пользователю значение можно получить

из файла `/usr/include/asm/unistd_64.h`. В данном случае, `257` — это системный вызов `openat`. Для преобразования номера системного вызова в его название можно использовать утилиту `ausyscall`:

```
$ ausyscall 257
openat
```

С помощью команды `ausyscall --dump` можно получить список номеров всех системных вызовов с их именами.

- `success=yes` — указывает на то, был ли системный вызов завершён успешно или возникла какая-то ошибка. В этом примере вызов был успешным.
- `exit=3` — код возврата, который вернул системный вызов. Это значение может отличаться для разных системных вызовов.
- `a0=ffffff9c a1=7ffd91e5b610 a2=0 a3=0` — в полях от `a0` до `a3` содержатся первые четыре аргумента системного вызова, закодированные в шестнадцатеричной системе счисления. Значения этих аргументов зависят от системного вызова и могут быть декодированы с помощью утилиты `ausearch`.
- `items=1` — количество вспомогательных записей типа `PATH`, которые следуют в журнале за данной записью системного вызова.
- `ppid=19085` — идентификатор родительского процесса (parent PID).
- `pid=20721` — идентификатор анализируемого процесса (PID).
- `auid=1667` — исходный идентификатор пользователя, использованный для входа в систему (audit id). Этот идентификатор наследуется каждым процессом, даже если идентификатор пользователя был изменён, например, при переключении учётных записей с помощью команды `su -`.
- `uid=1667` — идентификатор пользователя, запустившего анализируемый процесс.
- `gid=1667` — идентификатор группы пользователя, запустившего анализируемый процесс.
- `euid=1667` — действующий идентификатор пользователя, запустившего процесс. Определяет текущие полномочия процесса.
- `suid=1667` — установленный идентификатор пользователя, запустившего процесс. Используется для хранения начального значения `EUID`, задаваемого при запуске файла с установленным битом `set-user-ID`.
- `fsuid=1667` — идентификатор пользователя файловой системы, запустившего процесс.
- `egid=1667` — действующий идентификатор группы пользователя, запустившего процесс. Определяет текущие полномочия процесса.

- `sgid=1667` — установленный идентификатор группы пользователя, запустившего процесс. Используется для хранения начального значения `EGID`, задаваемого при запуске файла с установленным битом `set-group-ID`.
- `fsgid=1667` — идентификатор группы пользователя файловой системы, запустившего процесс.
- `tty=pts6` — терминал, с которого был запущен процесс. Типовые форматы значений для этого поля:
  - `ttyN` — физический терминал.
  - `pts/N` — терминал, который эмулируется какой-то программой, допустим, сервером SSH. Командные оболочки, запущенные в графической среде, будут являться pts-терминалами.
  - `(none)` — терминал отсутствует, обычно такое значение устанавливается для системных вызовов, выполняемых системными процессами.
- `ses=7` — идентификатор сессии, из которой был запущен процесс.
- `comm="cat"` — название команды, которая использовалась для запуска процесса.
- `exe="/usr/bin/cat"` — путь к исполняемому файлу, который использовался для запуска процесса.
- `subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023` — SELinux контекст анализируемого процесса.
- `key="fstab_read"` — определённый администратором ключ фильтрации, связанный с правилом, которое сгенерировало событие безопасности.

Для второй записи установлены следующие поля (описание поля `msg` пропущено, так как полностью совпадает с первой записью):

- `type=CWD` — тип `CWD` используется для записи о текущем рабочем каталоге, из которого был запущен анализируемый процесс, вызвавший системный вызов из первой записи. Цель этой записи состоит в том, чтобы зафиксировать местоположение текущего процесса на случай, если в запись с типом `PATH` будет записан относительный путь. Таким образом, можно будет восстановить абсолютный путь к выполненной команде.
- `cwd="/home/user"` — путь к каталогу, из которого был сделан системный вызов.

Для третьей записи установлены следующие поля (описание поля `msg` пропущено, так как полностью совпадает с первой записью):

- `type=PATH` — тип `PATH` используется для записей о каждом пути, который был передан системному вызову в качестве аргумента. В нашем случае, в системный вызов openat передавался только один путь — `/etc/fstab`. Для системных вызовов, принимающих несколько путей в качестве аргументов, подсистема аудита создала бы соответствующее количество записей данного типа.

- item=0 — порядковый номер этой записи среди общего числа записей типа PATH для данного события безопасности. Нумерация начинается с нуля, соответственно, в этом примере рассматривается первая запись.
- name="/etc/fstab" — путь к файлу или каталогу, который был передан системному вызову в качестве аргумента.
- inode=262528 — номер индексного дескриптора (inode), связанного с этим файлом или каталогом. Зная этот номер, можно определить путь к файлу или каталогу с помощью следующей команды:

```
$ find / -inum  262528 -print
/etc/fstab
```

- dev=fd:00 — младший (minor) и старший (major) номера устройства, которое содержит файл или каталог, записанный в этом событии.
- mode=0100644 — права доступа к файлу или каталогу, записанные в числовой форме, которые возвращаются функцией stat (man 2 stat) в поле st_mode. В данном случае 0100644 можно интерпретировать как -rw-r--r--, что означает, что пользователь root имеет право на чтение и запись, а остальные пользователи только на чтение.
- ouid=0 — идентификатор пользователя, который владеет заданным файлом или каталогом.
- ogid=0 — идентификатор группы, которая владеет заданным файлом или каталогом.
- rdev=00:00 — младший (minor) и старший (major) номера устройства для специальных файлов. В данном случае он не используется, потому что /etc/fstab является обычным файлом.
- obj=unconfined_u:object_r:etc_t:s0 — SELinux контекст файла или каталога на момент выполнения системного вызова.
- nametype=NORMAL — обозначает тип файлового объекта внутри контекста события. Подсистемой аудита используются следующие типы:
  - UNKNOWN — файловый объект не известен системе, например, его не существует.
  - NORMAL — обычный файловый объект. Как правило, это исполняемый файл или файл, у которого меняются атрибуты.
  - PARENT — файловый объект является родительским для одного из объектов, представленного в событии безопасности.
  - DELETE — файловый объект, удаляемый во время выполнения системного вызова.

  - CREATE — файловый объект, создаваемый во время выполнения системного вызова.

- cap_fp=0 — данные, относящиеся к настройке разрешённых возможностей файловой системы для файлового объекта.
- cap_fi=0 — данные, относящиеся к настройке унаследованных возможностей файловой системы для файлового объекта.
- cap_fe=0 — установка эффективного бита возможностей файловой системы для файлового объекта.
- cap_fver=0 — версия возможностей файловой системы для файлового объекта.

Для четвёртой записи установлены следующие поля (описание поля msg пропущено, так как полностью совпадает с первой записью):

- type=PROCTITLE — тип PROCTITLE указывает на то, что данная запись содержит полную команду запуска процесса, вызвавшего данное событие безопасности.
- proctitle=636174002F6574632F6673746162 — закодированная в шестнадцатеричной системе счисления полная команда запуска процесса. Утилита ausearch, запущенная с аргументом -i / --interpret автоматически преобразует закодированный текст в понятную пользователю строку, в нашем примере — в cat /etc/fstab.

Записи подсистемы аудита не ограничиваются перечисленными выше четырьмя типами, полный список типов доступен в конце этой главы.

Вероятнее всего вам никогда не потребуется работать с файлом журнала событий безопасности напрямую поскольку подсистема аудита включает в себя различные утилиты для поиска и обработки событий. Две основные, aureport и ausearch, будут рассмотрены в следующих разделах.

Утилита aureport

Команда aureport генерирует итоговые отчёты на основе зарегистрированных в журнале безопасности событий. Также aureport может принимать данные, поступающие на стандартный ввод (stdin), до тех пор пока на входе будут необработанные события аудита. За исключением основного сводного отчёта, все остальные отчёты содержат номер события в журнале аудита. Зная этот номер, вы можете посмотреть все данные по этому событию, используя команду ausearch -a <НОМЕР_СОБЫТИЯ>. Также существует возможность задать временной интервал, за который необходимо сгенерировать отчёт.

Опции командной строки утилиты aureport перечислены в таблице.

Таблица 22: Опции командной строки утилиты `aureport`

Аргумент	*Описание*
`-au, --auth`	Сгенерировать отчёт о попытках аутентификации.
`-a, --avc`	Сгенерировать отчёт о предоставлении разрешений SELinux AVC (Access Vector Cache).
`--comm`	Сгенерировать отчёт о выполнении команд.
`-c, --config`	Сгенерировать отчёт об изменениях конфигурации.
`-cr, --crypto`	Сгенерировать отчёт о событиях, связанных с криптографией.
`--debug`	Выводить искажённые события, которые были пропущены, в стандартный поток ошибок (`stderr`).
`--eoe-timeout <секунды>`	Устанавливает время ожидания разбора событий. Подробности доступны в описании директивы `end_of_event_timeout` конфигурационного файла `auditd.conf`. Значение, переданное в этом аргументе, имеет больший приоритет, чем значение, указанное в `auditd.conf`.
`-e, --event`	Сгенерировать отчёт о событиях.
`--escape <режим>`	Эта опция определяет, требуется ли экранирование вывода чтобы сделать его более безопасным для некоторых сценариев. Доступны следующие режимы экранирования: `raw`, `tty`, `shell` и `shell_quote`. Каждый режим включает правила экранирования из предыдущего режима и экранирует всё больше символов. Например, `shell` экранирует всё то, что экранируется в `tty` и добавляет экранирование дополнительных символов. Режим по умолчанию — `tty`.
`-f, --file`	Сгенерировать отчёт об операциях с файлами и Unix-сокетами.
`--failed`	Строить отчёт только на основе неудачных событий. По умолчанию показываются все события независимо от их статуса.
`-h, --host`	Сгенерировать отчёт о системе, который, среди прочего, включает события обновления ПО, аутентификации и т.п.
`--help`	Выдать справочную информацию об аргументах командой строки и завершить работу.
`-i, --interpret`	Включает преобразование числовых значений в текст. Например, идентификатор пользователя будет преобразован в его имя. Преобразование выполняется с использованием ресурсов текущего компьютера, на котором запущена команда `aureport`. Если вы переименовывали учётные записи или анализируете данные с другой системы, то вы можете получить ошибочные результаты.
`-if, --input < файл | каталог >`	Использовать указанный файл или каталог вместо системного журнала для построения отчёта. Это может быть полезно в случае анализа журналов на другом компьютере или если сохранилась только часть журнала.
`--input-logs`	Получить путь к журналу аудита для анализа из конфигурационного файла `audit.conf`. Применяется при автоматическом формировании отчётов через `cron`.
`--integrity`	Сгенерировать отчёт о событиях целостности.
`-k, --key`	Сгенерировать отчёт по ключевым словам в правилах аудита.
`-l, --login`	Сгенерировать отчёт о попытках входа в систему.
`-m, --mods`	Сгенерировать отчёт об изменениях учётных записей.
`-ma, --mac`	Сгенерировать отчёт об изменениях в системе мандатного доступа SELinux.
`-n, --anomaly`	Сгенерировать отчёт об аномальных событиях, сюда, в том числе, входят события о переходе сетевой карты в «неразборчивый» режим (promiscuous mode) и ошибки сегментирования (segmentation fault).
`--node <имя-узла>`	Использовать для выборки только события, поступившие с определённого узла. По умолчанию отчёт генерируется по всем узлам. Допускается указание имён нескольких узлов.
`-nc, --no-config`	Не включать в отчёт события с типом `CONFIG_CHANGE`. Это может быть полезным для отчёта по ключевым словам, поскольку многие правила аудита их используют — указание этой опции поможет избежать ложных срабатываний.
`-p, --pid`	Сгенерировать отчёт о процессах.
`-r, --response`	Сгенерировать отчёт о реагировании на аномальные события.
`-s, --syscall`	Сгенерировать отчёт о системных вызовах.
`--success`	Строить отчёт только на основе событий, завершившихся успешно.
`-t, --log`	Сгенерировать отчёт о временных периодах каждого файла журнала подсистемы аудита.
`--tty`	Сгенерировать отчёт о нажатиях клавиш в терминале.
`-te, --end [дата] [время]`	Учитывать только события, которые произошли раньше или во время указанной временной отметки. Формат даты зависит от ваших региональных настроек (см. описание переменной окружения `LC_TIME`). Если дата не указана, то используется значение `today`. Если время не указано, то используется значение `now`. Для указания времени используется 24-часовой формат. Ключевые слова перечислены после таблицы.
`-u, --user`	Сгенерировать отчёт о пользователях.
`-v, --version`	Вывести версию программы aureport и завершить работу.
`--virt`	Сгенерировать отчёт о событиях системы виртуализации.
`-x, --executable`	Сгенерировать отчёт об исполняемых файлах.

Ключевые слова агумента -te, –end [дата] [время]:

- `now` — сейчас.
- `recent` — 10 минут назад.
- `boot` — время за секунду до последней загрузки системы.
- `today` — сегодня.
- `yesterday` — 1 секунда после полуночи вчерашнего дня.
- `this-week` — 1 секунда после полуночи 0 (первого) дня текущей недели (определяется вашими региональными настройками).
- `week-ago` — 1 секунда после полуночи ровно 7 дней назад.

- `this-month` — 1 секунда после полуночи первого дня текущего месяца.
- `this-year` — 1 секунда после полуночи первого января текущего года.

Примеры использования утилиты aureport

Отчёт об изменениях учётных записей

Сгенерировать отчёт о событиях безопасности, связанных с изменением пользовательских учётных записей, за последние сутки:

```
$ sudo aureport --mods --start yesterday

Account Modifications Report
===============================================
# date time auid addr term exe acct success event
===============================================
1. 12/09/2024 18:11:56 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd devuser yes 2199
2. 12/09/2024 18:11:56 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd devuser yes 2200
3. 12/09/2024 18:11:56 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd devuser yes 2201
4. 12/09/2024 18:11:56 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd devuser yes 2202
5. 12/09/2024 18:11:56 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd ? yes 2203
6. 12/09/2024 18:12:04 1000 libvirt.msvsphere.test pts/11 /usr/bin/passwd devuser yes 2204
7. 12/09/2024 18:12:53 1000 libvirt.msvsphere.test pts/11 /usr/sbin/groupadd ? yes 2205
8. 12/09/2024 18:12:53 1000 libvirt.msvsphere.test pts/11 /usr/sbin/groupadd ? yes 2206
9. 12/09/2024 18:13:48 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd qauser yes 2207
10. 12/09/2024 18:13:48 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd qauser yes 2208
11. 12/09/2024 18:13:48 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd qauser yes 2209
12. 12/09/2024 18:13:48 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd qauser yes 2210
13. 12/09/2024 18:13:48 1000 libvirt.msvsphere.test pts/11 /usr/sbin/useradd ? yes 2211
14. 12/09/2024 18:13:59 1000 libvirt.msvsphere.test pts/11 /usr/bin/passwd qauser yes 2212
15. 12/09/2024 18:14:14 1000 libvirt.msvsphere.test pts/11 /usr/sbin/groupadd ? yes 2213
16. 12/09/2024 18:14:14 1000 libvirt.msvsphere.test pts/11 /usr/sbin/groupadd ? yes 2214
```

Результатом работы команды является таблица, в которой на каждой строке представлена информация о событии безопасности, а в столбцах следующая информация:

- `#` — порядковый номер строки в таблице;
- `date` — дата события;
- `time` — время события;
- `auid` — исходный идентификатор пользователя, инициировавшего событие. Вы также можете использовать аргумент `-i` для автоматического преобразования идентификатора в имя пользователя;
- `addr` — имя узла или его IP адрес;
- `term` — терминал, с которого была запущена команда, вызвавшая событие;
- `exe` — запущенная команда;
- `acct` — название учётной записи, для которой проводилось изменение;
- `success` — статус операции;
- `event` — идентификатор события.

Зная идентификатор события, можно посмотреть всю доступную информацию о нём с помощью команды `ausearch`.

```
$ sudo ausearch -a 2212
----
time->Mon Dec  9 18:13:59 2024
type=USER_CHAUTHTOK msg=audit(1733757239.466:2212): pid=589141 \
uid=0 auid=1000 ses=4 subj=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 \
msg='op=PAM:chauthtok grantors=pam_pwquality,pam_unix acct="qauser" \
exe="/usr/bin/passwd" hostname=libvirt.msvsphere.test addr=? terminal=pts/11 res=success'
```

В данном примере пользователь с идентификатором `1000` успешно изменил пароль пользователю `qauser` с помощью команды `passwd`.

Итоговый отчёт о входе пользователей в систему

Для некоторых данных поддерживается формирование итоговых отчётов с помощью аргумента `--summary`. Например, следующая команда генерирует итоговый отчёт о входе пользователей в систему за последние 7 дней.

```
$ sudo aureport -u -i --summary --start yesterday

User Summary Report
===========================
total  auid
===========================
1185  unset
36  virtadmin
27  gdm
24  vmuser
21  devuser
9  vmdev
7  qauser
1  root
```

В первом столбце указано суммарное количество зарегистрированных событий входа в систему, а во втором — исходный идентификатор пользователя, преобразованный в его имя с помощью аргумента `-i`.

6.4.3. Утилита ausearch

Утилита `ausearch` — это инструмент для поиска событий в журнале службы аудита на основе различных критерий. Опции командной строки утилиты `ausearch` перечислены в таблице.

Таблица 23: Опции командной строки утилиты `ausearch`

Аргумент	*Описание*
`-c, --comm <название_команды>`	Искать события, связанные с указанным именем команды.
`--arch <архитектура>`	Выполнить поиск событий для указанной архитектуры процессора. Вместо явного указания архитектуры вы также можете использовать константы `b32` для 32-битных архитектур и `b64` для 64-битных. Узнать архитектуру вашей системы можно с помощью команды `uname -m`.
`-a, --event <идентификатор_события>`	Выполнить поиск события с заданным идентификатором. Все сообщения от подсистемы аудита имеют поле вида `msg=audit(1731576783.677:1528): ...`, идентификатор события — это число после символа `:`, в данном примере — `1528`. Все записи, связанные с одним системным вызовом приложения, имеют одинаковый идентификатор. Следующий системный вызов, выполненный тем же приложением, уже будет иметь другой идентификатор, таким образом обеспечивается уникальность.
`--debug`	Выводить искажённые события, которые были пропущены, в стандартный поток ошибок (stderr).

продолжение на следующей странице

Таблица 23 – продолжение с предыдущей страницы

Аргумент	*Описание*
-w, --word <слово>	Выполнить поиск событий, у которых значение поля полностью совпадает с указанным словом. Поддерживаются следующие поля: имя файла, имя компьютера, терминал, SELinux контекст.
-x, --executable <программа>	Выполнить поиск событий с заданным именем исполняемой программы.
-vm, --vm-name <название_гостевой_системы>	Выполнить поиск событий, связанных с заданным названием гостевой системы (виртуальной машины).
--checkpoint <файл_контрольной_точки>	Использовать файл контрольной точки для сохранения состояния, чтобы при последующих вызовах ausearch выводились только события, которые не отображались ранее. События подсистемы аудита могут состоять из одной или нескольких записей. При их обработке ausearch определяет событие как завершённое или незавершённое. Завершённое — это либо событие с одной записью, либо событие, которое произошло на две секунды (см. описание опции --eoe-timeout) раньше по сравнению с текущим обрабатываемым событием (см. описание директивы end_of_event_timeout в конфигурационном файле auditd.conf). При использовании опции --checkpoint в файл_контрольной_точки записывается последнее завершённое событие, а также номер устройства и номер индексного дескриптора (inode) файла журнала, из которого было получено это событие. При следующем вызове ausearch загрузит эти данные из файла контрольной точки и будет игнорировать все события из журналов до тех пор, пока не будет обнаружено совпадение с контрольной точкой. После этого утилита начнёт выводить завершённые события. Если указанный в контрольной точке файл или последнее событие не будут найдены, то ausearch завершит свою работу с ошибкой (см. таблицу кодов возврата ниже).
--eoe-timeout <секунды>	Устанавливает количество секунд, после которого событие считается завершённым при анализе потока журнала событий. Подробности смотрите в описании директивы end_of_event_timeout конфигурационного файла auditd.conf. Установка этой опции переопределит значение, указанное в файле auditd.conf.
-e, --exit <код>	Выполнить поиск событий на основе указанного кода возврата из системного вызова или кода ошибки errno (man errno).
--escape <режим>	Эта опция определяет, требуется ли экранирование вывода чтобы сделать его более безопасным для некоторых сценариев. Доступны следующие режимы экранирования: raw, tty, shell и shell_quote. Каждый режим включает правила экранирования из предыдущего режима и экранирует всё больше символов. Например, shell экранирует всё то, что экранируется в tty и добавляет экранирование дополнительных символов. Режим по умолчанию — tty.
--extra-keys	Если format установлен в csv, эта опция добавит последний столбец с информацией о ключе, если он задан для события. Это будет применимо только к записям типа SYSCALL, которые были записаны в результате обработки правила аудита, определяющего ключ.
--extra-labels	Если format установлен в csv, эта опция добавит столбцы с SELinux метками субъекта и объекта если они определены.
--extra-obj2	Если format установлен в csv, эта опция добавит информацию о втором объекте, если он указан в записи о событии. Второй объект иногда является частью записи, например, при переименовывании файла или монтировании устройства.
--extra-time	Если format установлен в csv, эта опция добавит дополнительные столбцы с разобранным по полям временем и датой: YEAR, MONTH, DAY, WEEKDAY, HOUR, MILLI и GMT_OFFSET.
-f, --file <имя_файла>	Выполнить поиск событий, связанных с указанным именем файла. Эта опция будет применима как к обычным файлам, так и к Unix-сокетами (AF_UNIX/AF_LOCAL).
--format <формат>	Определяет формат вывода сообщений, соответствующих критериям поиска. Допустимые значения параметра перечислены после таблицы.
-ga, --gid-all <идентификатор_группы>	Выполнить поиск событий, у которых либо действующий идентификатор группы, либо идентификатор группы совпадает с заданным идентификатором группы.
-ge, --gid-effective <идентификатор_группы>	Выполнить поиск событий, у которых действующий идентификатор группы соответствует заданному идентификатору группы. Также в качестве параметра можно использовать название группы.
-gi, --gid <идентификатор_группы>	Выполнить поиск событий, у которых идентификатор группы соответствует заданному идентификатору группы. Также в качестве параметра можно использовать название группы.
--help	Выдать справочную информацию об аргументах командой строки и завершить работу.
-hn, --host <имя_узла>	Выполнить поиск событий, связанных с заданным именем узла. Имя узла может быть именем компьютера, полным именем компьютера (FQDN) или IP-адресом. Преобразование IP-адресов в доменные имена не выполняется. Обычно этот критерий поиска коррелирует с полями записей addr или host. Также смотрите описание опции --node, которая выполняет поиск по полю node.
-i, --interpret	Включает преобразование числовых значений в текст. Например, идентификатор пользователя будет преобразован в его имя. Преобразование выполняется с использованием ресурсов текущего компьютера, на котором запущена команда aureport. Если вы переименовывали учётные записи или анализируете данные с другой системы, то вы можете получить ошибочные результаты.
-if, --input <файл | каталог>	Использовать указанный файл или каталог вместо системного журнала для построения отчёта. Это может быть полезно в случае анализа журналов на другом компьютере или если сохранилась только часть журнала.
--input-logs	Получить путь к журналу аудита для анализа из конфигурационного файла audit.conf. Применяется при автоматическом формировании отчётов через cron.
--just-one	Остановить поиск после выдачи первого события, соответствующего критериям поиска.
-k, --key <ключ>	Выполнить поиск событий, связанных с указанным ключом. Подробности доступны в описании опции -k команды auditctl.
-l, --line-buffered	Сбрасывать буфер вывода после каждой строки. Обычно используется когда стандартный вывод перенаправляется через канал и буферизация вывода является нежелательной. Использование этой опции может привести к снижению производительности.
-m, --message <тип | список_типов>	Выполнить поиск записей о событиях с указанным типом. Поддерживается указание нескольких типов через запятую без пробелов или с помощью отдельных параметров -m. Вы также можете использовать не существующий реально тип ALL, который включает в себя события всех типов. Вы можете посмотреть список всех поддерживаемых типов, если запустите команду ausearch -m без указания типа.
-n, --node <имя_узла | список_имён_узлов>	Выполнить поиск событий, исходящих от определённого компьютера (узла). Допускается перечисление нескольких узлов через запятую, для вывода события достаточно совпадения с одним из перечисленных имён. Поиск выполняется по полю записи node. Также смотрите описание директивы --host, которая выполняет поиск событий, связанных с именем или IP-адресом узла.

продолжение на следующей странице

Таблица 23 – продолжение с предыдущей страницы

Аргумент	*Описание*
-o, --object <контекст_SELinux>	Выполнить поиск событий, связанных с объектами, которым присвоен указанный контекст SELinux. Поиск выполняется по полю obj.
-p, --pid <pid>	Выполнить поиск событий, связанных с заданным идентификатором процесса (pid).
-pp, --ppid <pid_родительского_процесса>	Выполнить поиск событий, связанных с заданным идентификатором родительского процесса (parent pid).
-r, --raw	Не применять какое-либо форматирование к выводу. Это полезно для извлечения найденных записей в файл, с которым смогут продолжать работать команды подсистемы аудита.
-sc, --syscall <системный_вызов>	Выполнить поиск событий, связанных с указанным системным вызовом. Вы можете указать либо название, либо номер системного вызова. Если вы указываете название, то ausearch определит номер вызова на основании таблицы системных вызовов для архитектуры компьютера, на котором запущена команда.
-se, --context <контекст_SELinux>	Выполнить поиск событий, связанных с объектами или субъектами, которым присвоен указанный контекст SELinux. Поиск выполняется по полям obj и subj.
--session <сессия>	Выполнить поиск событий, связанных с заданным идентификатором пользовательской сессии. Этот атрибут устанавливается при входе пользователя в систему и позволяет связать процесс с определённым сеансом пользователя.
-su, --subject <контекст_SELinux>	Выполнить поиск событий, связанных с субъектами, которым присвоен указанный контекст SELinux. Поиск выполняется по полю subj.
-sv, --success <статус>	Выполнить поиск событий, завершившихся с заданным статусом. Допустимые значения: yes (успешно) и no (неудачно).
-ul, --loginuid <идентификатор_пользователя>	Выполнить поиск событий, у которых исходный идентификатор пользователя (auid) соответствует указанному идентификатору пользователя.
-te, --end [дата] [время]	Учитывать только события, которые произошли раньше или во время указанной временной отметки. Формат даты зависит от ваших региональных настроек (см. описание переменной окружения LC_TIME). Если дата не указана, то используется значение today. Если время не указано, то используется значение now. Для указания времени используется 24-часовой формат. Ключевые слова для параметра перечислены после таблицы.
-ts, --start [дата] [время]	Учитывать только события, которые произошли позже или во время указанной временной отметки. Формат даты зависит от ваших региональных настроек (см. описание переменной окружения LC_TIME). Если дата не указана, то используется значение today. Если время не указано, то используется значение now. Для указания времени используется 24-часовой формат. Ключевые слова для параметра перечислены после таблицы.
-tm, --terminal <терминал>	Выполнить поиск событий, связанных с заданным терминалом. Некоторые службы, такие как cron и atd, используют имя службы как имя терминала.
-ua, --uid-all <идентификатор_пользователя>	Выполнить поиск событий, у которых либо идентификатор пользователя (uid), либо действующий идентификатор пользователя (euid), либо исходный идентификатор пользователя (auid) соответствуют указанному идентификатору пользователя.
-ue, --uid-effective <идентификатор_пользователя>	Выполнить поиск событий, у которых действующий идентификатор пользователя (euid) соответствует указанному идентификатору пользователя.
-ui, --uid <идентификатор_пользователя>	Выполнить поиск событий, у которых идентификатор пользователя (uid) соответствует указанному идентификатору пользователя.
-uu, --uuid <идентификатор_гостевой_системы>	Выполнить поиск событий, связанных с заданным идентификатором гостевой системы (UUID).
-v, --version	Вывести версию утилиты ausearch и завершить работу.

Допустимые значения параметра –format <формат>:

- raw — смотрите описание к опции --raw.
- default — формат, в котором вы получаете вывод если опция --format не задана: сначала идёт строка-разделитель, затем — временная отметка события, после — все записи, связанные с этим событием.
- interpret — смотрите описание к опции --interpret.
- csv — выводит результаты поиска в виде нормализованных событий в формате значений, разделённых запятой (CSV) — этот формат подходит для импорта в аналитические программы.
- text — превращает каждое событие в предложение на английском языке, которое легче понять, но при этом теряются различные детали. В большинстве случаев это считается нормальным, поскольку исходное событие по прежнему содержит полный объём информации.

Ключевые слова для параметра -te, –end [дата] [время]:

- now — сейчас.

- `recent` — 10 минут назад.
- `boot` — время за секунду до последней загрузки системы.
- `today` — сегодня.
- `yesterday` — 1 секунда после полуночи вчерашнего дня.
- `this-week` — 1 секунда после полуночи 0 (первого) дня текущей недели (определяется вашими региональными настройками).
- `week-ago` — 1 секунда после полуночи ровно 7 дней назад.
- `this-month` — 1 секунда после полуночи первого дня текущего месяца.
- `this-year` — секунда после полуночи первого января текущего года.

Ключевые слова для параметра -ts, –start [дата] [время]:

- `now` — сейчас.
- `recent` — 10 минут назад.
- `boot` — время за секунду до последней загрузки системы.
- `today` — сегодня.
- `yesterday` — 1 секунда после полуночи вчерашнего дня.
- `this-week` — 1 секунда после полуночи 0 (первого) дня текущей недели (определяется вашими региональными настройками).
- `week-ago` — 1 секунда после полуночи ровно 7 дней назад.
- `this-month` — 1 секунда после полуночи первого дня текущего месяца.
- `this-year` — 1 секунда после полуночи первого января текущего года.
- `checkpoint` — `ausearch` будет использовать временную отметку из файла контрольной точки игнорируя при этом идентификатор последнего завершённого события, номер устройства и номер индексного дескриптора (`inode`) файла журнала (см. описание опции `--checkpoint`). По сути это действие для восстановления, если вызов `ausearch --checkpoint` завершился с кодом возврата `10`, `11` или `12` (см. таблицу кодов возврата ниже).

  Эту опцию можно использовать в сценариях командной оболочки следующим образом:

```
ausearch --checkpoint /etc/audit/auditd_checkpoint.txt -i
_au_status=$?
if test ${_au_status} eq 10 -o ${_au_status} eq 11 -o ${_au_status} eq 12
then
  ausearch --checkpoint /etc/audit/auditd_checkpoint.txt --start checkpoint -i
fi
```

Коды возврата утилиты `ausearch` перечислены в таблице.

Таблица 24: Коды возврата утилиты `ausearch`

Код возврата	*Описание*
0	Операция выполнена успешно.
1	Если не найдено событий, соответствующих условию, или возникла ошибка при обработке опций командной строки, или возникли незначительные ошибки доступа/чтения файлов.
10	В файле контрольной точки обнаружены неверные данные.
11	Ошибка обработки контрольной точки.
12	Событие из контрольной точки не найдено в соответствующем файле журнала.

Примеры использования утилиты ausearch

Поиск по типу события

Отобразить события безопасности за последние сутки, связанные с входом пользователя в систему:

```
$ sudo ausearch -m USER_LOGIN,LOGIN --start yesterday
----
time->Tue Dec 17 14:38:08 2024
type=LOGIN msg=audit(1734435488.883:195): pid=2132 uid=0 \
subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 old-auid=4294967295 \
auid=1004 tty=(none) old-ses=4294967295 ses=4 res=1
----
time->Tue Dec 17 14:38:08 2024
type=LOGIN msg=audit(1734435488.907:201): pid=2146 uid=0 \
subj=system_u:system_r:init_t:s0 old-auid=4294967295 auid=1004 \
tty=(none) old-ses=4294967295 ses=5 res=1
----
time->Wed Dec 18 15:49:57 2024
type=LOGIN msg=audit(1734526197.764:621): pid=256120 uid=0 \
subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 old-auid=4294967295 \
auid=1000 tty=(none) old-ses=4294967295 ses=6 res=1
----
time->Wed Dec 18 15:49:57 2024
type=USER_LOGIN msg=audit(1734526197.821:626): pid=256120 uid=0 \
auid=1000 ses=6 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 \
msg='op=login id=1000 exe="/usr/sbin/sshd" hostname=? addr=192.168.1.4 terminal=/dev/pts/3␣
↪res=success'
```

С помощью аргумента `-i` / `--interpret` можно преобразовать числовые идентификаторы пользователей в имена:

```
$ sudo ausearch -m USER_LOGIN,LOGIN --start yesterday -i
----
type=LOGIN msg=audit(12/17/2024 14:38:08.883:195) : \
pid=2132 uid=root subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
old-auid=unset auid=virtadmin tty=(none) old-ses=4294967295 ses=4 res=yes
----
type=LOGIN msg=audit(12/17/2024 14:38:08.907:201) : \
pid=2146 uid=root subj=system_u:system_r:init_t:s0 old-auid=unset \
auid=virtadmin tty=(none) old-ses=4294967295 ses=5 res=yes
----
type=LOGIN msg=audit(12/18/2024 15:49:57.764:621) : \
pid=256120 uid=root subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 \
old-auid=unset auid=virtuser tty=(none) old-ses=4294967295 ses=6 res=yes
----
type=USER_LOGIN msg=audit(12/18/2024 15:49:57.821:626) : \
pid=256120 uid=root auid=virtuser ses=6 subj=system_u:system_r:sshd_t:s0-s0:c0.c1023 \
msg='op=login id=virtuser exe=/usr/sbin/sshd hostname=? addr=192.168.1.4 terminal=/dev/pts/3␣
↪res=success'
----
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
type=LOGIN msg=audit(12/18/2024 15:50:36.490:657) : pid=256436 \
uid=root subj=system_u:system_r:init_t:s0 old-auid=unset auid=gdm \
tty=(none) old-ses=4294967295 ses=7 res=yes
```

Поиск по идентификатору пользователя

Отобразить события безопасности за сегодняшний день, связанные с определённым идентификатором или именем пользователя:

```
$ sudo ausearch --uid-all virtadmin --start today
----
time->Wed Dec 18 15:50:16 2024
type=USER_AUTH msg=audit(1734526216.077:641): pid=256200 uid=0 \
auid=1004 ses=4 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
msg='op=PAM:authentication grantors=pam_usertype,pam_localuser, \
pam_unix,pam_gnome_keyring acct="virtadmin" exe="/usr/libexec/gdm-session-worker" \
hostname=libvirt.msvsphere.test addr=? terminal=/dev/tty1 res=success'
----
time->Wed Dec 18 15:50:16 2024
type=USER_ACCT msg=audit(1734526216.079:642): pid=256200 \
uid=0 auid=1004 ses=4 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
msg='op=PAM:accounting grantors=pam_unix,pam_localuser acct="virtadmin" \
exe="/usr/libexec/gdm-session-worker" hostname=libvirt.msvsphere.test \
addr=? terminal=/dev/tty1 res=success'
----
time->Wed Dec 18 15:50:16 2024
type=CRED_REFR msg=audit(1734526216.080:643): pid=256200 \
uid=0 auid=1004 ses=4 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
msg='op=PAM:setcred grantors=pam_localuser,pam_unix,pam_gnome_keyring \
acct="virtadmin" exe="/usr/libexec/gdm-session-worker" hostname=libvirt.msvsphere.test addr=?␣
↪terminal=/dev/tty1 res=success'
----
time->Wed Dec 18 15:50:36 2024
type=USER_END msg=audit(1734526236.347:648): pid=2132 \
uid=0 auid=1004 ses=4 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
msg='op=PAM:session_close grantors=pam_selinux,pam_loginuid,pam_selinux,pam_keyinit,\
pam_namespace,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_gnome_keyring,pam_umask \
acct="virtadmin" exe="/usr/libexec/gdm-session-worker" hostname=libvirt.msvsphere.test \
addr=? terminal=/dev/tty2 res=success'
----
time->Wed Dec 18 15:50:36 2024
type=CRED_DISP msg=audit(1734526236.347:649): pid=2132 uid=0 auid=1004 \
ses=4 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 msg='op=PAM:setcred \
grantors=pam_localuser,pam_unix,pam_gnome_keyring acct="virtadmin" \
exe="/usr/libexec/gdm-session-worker" hostname=libvirt.msvsphere.test \
addr=? terminal=/dev/tty2 res=success'
----
time->Wed Dec 18 15:50:46 2024
type=LOGIN msg=audit(1734526246.330:677): pid=256830 uid=0 \
subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 old-auid=4294967295 \
auid=1004 tty=(none) old-ses=4294967295 ses=8 res=1
----
time->Wed Dec 18 15:50:46 2024
type=PROCTITLE msg=audit(1734526246.330:677):␣
↪proctitle=67646D2D73657373696F6E2D776F726B6572205B70616\
D2F67646D2D70617373776F72645D
type=SYSCALL msg=audit(1734526246.330:677): arch=c000003e \
syscall=1 success=yes exit=4 a0=9 a1=7ffd6aa865a0 a2=4 a3=3ec \
items=0 ppid=1020 pid=256830 auid=1004 uid=0 gid=1004 euid=0 suid=0 \
fsuid=0 egid=1004 sgid=1004 fsgid=1004 tty=(none) ses=8 comm="gdm-session-wor" \
exe="/usr/libexec/gdm-session-worker" subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 key=(null)
----
time->Wed Dec 18 15:50:46 2024
type=USER_ROLE_CHANGE msg=audit(1734526246.330:678): pid=256830 \
uid=0 auid=1004 ses=8 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
msg='op=pam_selinux default-context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 \
selected-context=unconfined_u:unconfined_r: \
unconfined_t:s0-s0:c0.c1023 exe="/usr/libexec/gdm-session-worker" \
hostname=libvirt.msvsphere.test addr=? terminal=/dev/tty2 res=success'
----
time->Wed Dec 18 15:50:46 2024
type=USER_START msg=audit(1734526246.352:679): pid=256830 \
uid=0 auid=1004 ses=8 subj=system_u:system_r:xdm_t:s0-s0:c0.c1023 \
msg='op=PAM:session_open grantors=pam_selinux,pam_loginuid,pam_selinux,pam_keyinit, \
pam_namespace,pam_keyinit,pam_limits,pam_systemd,pam_unix,pam_gnome_keyring, \
pam_umask acct="virtadmin" exe="/usr/libexec/gdm-session-worker" \
hostname=libvirt.msvsphere.test addr=? terminal=/dev/tty2 res=success'
```

Выгрузка данных из журнала

Утилиту `ausearch` можно использовать для выгрузки данных из журнала безопасности для их последующей обработки.

Например, можно экспортировать результаты в формате CSV:

```
$ sudo ausearch --uid-all virtadmin --start today --format csv
NODE,EVENT,DATE,TIME,SERIAL_NUM,EVENT_KIND,SESSION,SUBJ_PRIME,SUBJ_SEC, \
SUBJ_KIND,ACTION,RESULT,OBJ_PRIME,OBJ_SEC,OBJ_KIND,HOW
,USER_AUTH,12/18/2024,15:50:16,641,user-login,4,virtadmin,root, \
privileged-acct,authenticated,success,virtadmin,,user-session,/usr/libexec/gdm-session-worker
,USER_ACCT,12/18/2024,15:50:16,642,user-login,4,virtadmin,root, \
privileged-acct,was-authorized,success,virtadmin,,user-session,/usr/libexec/gdm-session-worker
,CRED_REFR,12/18/2024,15:50:16,643,user-login,4,virtadmin,root, \
privileged-acct,refreshed-credentials,success,virtadmin,,user-session,/usr/libexec/gdm-session-
↪worker
,USER_END,12/18/2024,15:50:36,648,user-login,4,virtadmin,root, \
privileged-acct,ended-session,success,/dev/tty2,,user-session,/usr/libexec/gdm-session-worker
,CRED_DISP,12/18/2024,15:50:36,649,user-login,4,virtadmin,root, \
privileged-acct,disposed-credentials,success,virtadmin,,user-session,/usr/libexec/gdm-session-
↪worker
,LOGIN,12/18/2024,15:50:46,677,user-login,8,system,root, \
privileged-acct,changed-login-id-to,success,virtadmin,,user-session,
,SYSCALL,12/18/2024,15:50:46,677,audit-rule,8,virtadmin,root, \
privileged-acct,triggered-unknown-audit-rule,success,,,admin-defined-rule,/usr/libexec/gdm-
↪session-worker
,USER_ROLE_CHANGE,12/18/2024,15:50:46,678,mac,8,virtadmin,root, \
privileged-acct,changed-role-to,success,unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023,\
,user-session,/usr/libexec/gdm-session-worker
,USER_START,12/18/2024,15:50:46,679,user-login,8,virtadmin,root, \
privileged-acct,started-session,success,/dev/tty2,,user-session,/usr/libexec/gdm-session-worker
```

Также можно экспортировать фрагменты журнала в «сыром» (raw) формате для их последующей обработки с помощью `ausearch` или других утилит для работы с журналами событий безопасности.

Например, следующая команда экспортирует все события входа пользователей в систему за последнюю неделю в файл `weekly-logins.log`:

```
$ sudo ausearch -m LOGIN,USER_LOGIN --start week-ago --raw > weekly-logins.log
```

С полученным файлом можно работать как с обычным журналом службы аудита, например выполнить фильтрацию по определённому пользователю и преобразовать вывод в CSV формат:

```
$ ausearch --uid-all 1004 --format csv --input weekly-logins.log
NODE,EVENT,DATE,TIME,SERIAL_NUM,EVENT_KIND,SESSION,SUBJ_PRIME,SUBJ_SEC,SUBJ_KIND,
ACTION,RESULT,OBJ_PRIME,OBJ_SEC,OBJ_KIND,HOW
,LOGIN,12/16/2024,12:03:51,1269,user-login,10,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/16/2024,12:03:51,1275,user-login,11,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/16/2024,21:11:17,173,user-login,2,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/16/2024,21:11:17,179,user-login,3,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/17/2024,14:38:08,195,user-login,4,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/17/2024,14:38:08,201,user-login,5,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
,LOGIN,12/18/2024,15:50:46,677,user-login,8,system,root,privileged-acct,
changed-login-id-to,success,virtadmin,,user-session,
```

Резервное копирование журналов событий безопасности

Поскольку журналы событий безопасности хранятся в виде файлов в каталоге `/var/log/audit`, для создания их резервной копии может применяться любой инструмент резервного копирования, поддерживающий работу с файлами. В состав операционной системы включены утилиты `tar`, `rsync` и система резервного копирования `bacula`.

Простой пример создания локальной резервной копии с помощью команды `tar`:

```
$ tar -cjpvf "auditd-logs.$(date --iso-8601).tar.bz2" /var/log/audit/audit.log*
```

В результате выполнения команды в текущем каталоге будет создан файл `auditd-logs.YYYY-MM-DD.tar.bz2` где `YYYY` — год, `MM` — месяц и `DD` — сегодняшнее число. В архив будут помещены все файлы из каталога `/var/log/audit`, соответствующие шаблону `audit.log*`.

Однако, при таком подходе есть некоторый риск получить неконсистентную копию файла текущего журнала поскольку, теоретически, в момент копирования в этот файл может происходить запись.

Более надёжным решением будет подать службе аудита сигнал на выполнение принудительной ротации файлов журнала. Получив такой сигнал, служба аудита выполнит следующие действия:

- переименует все имеющиеся копии журнала увеличив число в расширении файла на единицу: `audit.log.1` будет переименован в `audit.log.2`, `audit.log.2` — в `audit.log.3` и т.д.;
- остановит запись в текущий файл журнала `audit.log`;
- переименует текущий файл журнала в `audit.log.1`;
- создаст новый файл журнала `audit.log` и продолжит запись событий уже в него.

Таким образом будет обеспечена консистентность текущего файла журнала при его резервном копировании.

С учётом вышеизложенного, более правильным будет использовать следующий набор команд, оформленный в виде сценария командной оболочки, для создания резервной копии:

```
#!/bin/bash

# завершить работу программы в случае возникновения ошибки
set -e

# сменить текущий каталог на /srv/backup
cd /srv/backup

# выполнить принудительную ротацию журналов службы auditd
auditctl --signal rotate

# создать архив со всеми файлами, соответствующими шаблону
tar -cjpvf "auditd-logs.$(date --iso-8601).tar.bz2" /var/log/audit/audit.log.*
```

Автоматизировать создание резервных копий можно с помощью службы периодического выполнения заданий `cron` или таймеров `systemd`. Например, для создания ежедневных архивов вы можете сохранить указанный выше сценарий в файл в каталоге `/etc/cron.daily` (например, `audit-logs-backup.sh`) и сделать его исполняемым:

```
$ sudo chmod +x /etc/cron.daily/audit-logs-backup.sh
```

После этого служба cron будет автоматически выполнять сценарий ежедневно.

Для восстановления можно использовать следующую команду (замените `auditd-logs.2024-12-28.tar.bz2` на реальное имя файла):

```
$ tar -C / -xjpvf auditd-logs.2024-12-28.tar.bz2
```

Контроль целостности сведений о событиях безопасности

Целостность сведений о событиях безопасности обеспечивается на уровне ограничения прав доступа в операционной системе: каталог с журналами безопасности `/var/log/audit` и текущий файл журнала `/var/log/audit/audit.log` доступны для чтения и записи только привилегированному пользователю `root` и службе `auditd`. В свою очередь, предыдущие файлы журналов доступны только для чтения привилегированному пользователю `root`.

Для повышения защищённости системы рекомендуется использовать дополнительные средства контроля целостности, например, утилиту `aide`, которая входит в состав дистрибутива МСВСфера ОС. Документация по установке и настройке `aide` доступна в разделе «Контроль целостности».

При изменении прав доступа, владельца и других атрибутов каталога `/var/log/audit` или файла журнала событий безопасности `/var/log/audit/audit.log` команда `aide` выдаст соответствующее предупреждение о нарушении целостности в отчёте:

```
aide --check
Start timestamp: 2025-01-20 06:48:56 +0000 (AIDE 0.16)
AIDE found differences between database and filesystem!!

Summary:
  Total number of entries:      37822
  Added entries:                0
  Removed entries:              0
  Changed entries:              2

---------------------------------------------------
Changed entries:
---------------------------------------------------

d   p..      A.. : /var/log/audit
f   ..g      ... : /var/log/audit/audit.log

---------------------------------------------------
Detailed information about changes:
---------------------------------------------------

Directory: /var/log/audit
  Perm     : drwx------                       | drwxr-xr-x
  ACL      : A: user::rwx                     | A: user::rwx
           A: group::---                    | A: group::r-x
           A: other::---                    | A: other::r-x

File: /var/log/audit/audit.log
  Gid      : 0                                | 1000


---------------------------------------------------
The attributes of the (uncompressed) database(s):
---------------------------------------------------

/var/lib/aide/aide.db.gz
  MD5      : AABMsEbRMfWShAvA8Yl8kg==
  SHA1     : FssoJmeKJvo7VMc79ZuV1bGgNwI=
  RMD160   : OXPu+xyPaeV6I2c8AOrxwZx8EKA=
  TIGER    : Cez1vaIx/3koN5MbQOwktp/D247COpTa
  SHA256   : B8eWuoZeZ/9PsRrFJIV7lmrXBHo5DPbD
             qaBEO4iea1E=
  SHA512   : xmC1vT9hx9jXmX8NZDbzwUpsaldBbUPj
             F95IEPWxaJn8I3PQnR4G2fZZHnz6mG9G
             OW222CS6V3s2u25O5SqiTQ==


End timestamp: 2025-01-20 06:48:58 +0000 (run time: 0m 2s)
```

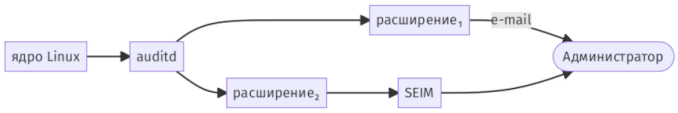
Оповещение о событиях безопасности

Основной задачей подсистемы аудита является регистрация событий безопасности — автоматическое отслеживаниие таких событий и занесение их в системный журнал.

Оповещение о возникновении событий безопасности и, при необходимости, автоматизированное реагирование на такие события реализуется с помощью дополнительных инструментов. В данном разделе рассмотрены несколько вариантов решения данной задачи.

Разработка расширений для службы аудита

Служба `auditd` имеет встроенный механизм расширений (плагинов), за счёт которого можно значительно расширить функциональность системы.



Например, с помощью расширений можно в реальном времени фильтровать определённые события, оповещать администраторов, загружать информацию о событиях безопасности в другие системы и предпринимать любые другие запрограммированные действия.

Расширение службы аудита представляет собой программу, реализованную на любом языке программирования, которая соответствует следующим требованиям:

- непрерывно получает поток событий на стандартный поток ввода (`stdin`), формат получаемых данных идентичен тому, в котором события записываются в системный журнал событий безопасности;
- обрабатывает системные сигналы `HUP` (обновить настройки из конфигурационного файла) и `TERM` (завершить работу);
- обрабатывает события максимально быстро, не допуская блокировок и переполнения очереди событий, ожидающих отправки со стороны службы `auditd`;
- предоставляет конфигурационный файл, который информирует службу аудита о способе запуска программы-расширения.

В состав операционной системы МСВСфера 9 сертифицированная редакция входят библиотеки для языков программирования Си (пакет *audit-libs-devel*) и Python (пакет *python3-audit*), которые предоставляют набор готовых функций для разбора поступающих данных. В случае с остальными языками программирования вам потребуется разработать соответствующий парсер самостоятельно.

Рассмотрим подход к созданию расширений для службы аудита на примере создания простого плагина на языке программирования Python, отправляющего уведомления о входе пользователя в систему по электронной почте.

Исходный код расширения с пояснениями:

```
#!/usr/bin/python3 -I
# аргумент -I указывает на необходимость запуска в изолированном режиме: не
# выполняется поиск модулей в каталоге с программой и в пользовательском
# домашнем каталоге, так же игнорируются все переменные окружения PYTHON*.

# импорт необходимых модулей из стандартной библиотеки Python
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
import datetime
import email.message
import pprint
import signal
import smtplib
import sys

# модуль auparse поставляется в RPM пакете python3-audit и предоставляет функции
# для разбора событий, поступающих от службы аудита
import auparse

# исходящий почтовый адрес для отправляемых уведомлений
FROM_EMAIL = 'audit-plugin-notify@localhost'
# список почтовых адресов, на которые необходимо отправлять уведомления
ADMIN_EMAILS = ['secadmin@localhost']


def reload_config():
    """
    Функция-заглушка, внутри которой может быть реализована логика обработки
    сигнала HUP - обновление настроек из конфигурационного файла.
    """
    pass


def notify_user_login(event: dict):
    """
    Функция, которая отправляет электронное письмо с уведомлением о входе
    пользователя в систему.
    """
    # поле "subj" содержит информацию от подсистемы SELinux, благодаря типу
    # можно определить каким образом пользователь вошёл в систему
    se_subj = event['fields'].get('subj', '').split(':')
    if len(se_subj) < 3:
        return
    se_type = se_subj[2]
    if se_type == 'sshd_t':
        # удалённый вход в систему по протоколу SSH
        login_type = 'SSH'
    elif se_type == 'local_login_t':
        # локальный вход в систему через текстовый терминал
        login_type = 'terminal'
    elif se_type == 'xdm_t':
        # вход в систему в графическом режиме
        login_type = 'graphical console'
    else:
        # игнорировать события остального типа, как правило это системные
        # события типа "init_t", которые отображают активность внутренних
        # компонентов системы.
        return
    # сформировать почтовое сообщение и отправить его через локальный сервер
    # электронной почты. В данном случае локальный сервис принимает почту от
    # локальных пользователей без дополнительной аутентификации. Библиотека
    # smptlib также поддерживает аутентификацию и подключение по защищённому
    # протоколу SMTPs, так что при необходимости код можно модифицировать для
    # использования любого другого SMTP сервиса для отправки.
    msg_subj = f'User {event["fields"]["auid"]} logged in via {login_type}'
    with smtplib.SMTP('localhost') as smtp:
        msg = email.message.EmailMessage()
        msg['Subject'] = msg_subj
        msg['To'] = ', '.join(ADMIN_EMAILS)
        msg['From'] = FROM_EMAIL
        msg.set_content(pprint.pformat(event))
        smtp.send_message(msg)


def parse_record(parser: auparse.AuParser):
    """
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    Функция, которая разбирает отдельную запись события безопасности.
    """
    # метод parser.get_timestamp возвращает объект, который содержит временную
    # метку записи, имя узла и идентификатор события
    event = parser.get_timestamp()
    # преобразовать данные из записи в словарь (хеш-таблицу) чтобы облегчить
    # их последующую обработку
    data = {
        'type': parser.get_type_name(),
        'event': {
            'host': event.host,
            'ts': datetime.datetime.fromtimestamp(event.sec),
            'serial': event.serial
        },
        'fields': {}
    }
    # выполнить обход всех полей записи и записать информацию в словарь
    # data["fields"]. Метод parser.first_field "передвигает курсор" на первое
    # поле записи
    parser.first_field()
    while True:
        data['fields'][parser.get_field_name()] = parser.interpret_field()
        # метод parser.next_field "передвигает курсор" на следующее поле записи,
        # если полей больше нет, будет возвращено значение False
        if not parser.next_field():
            break
    # вызвать функцию notify_user_login если тип записи - LOGIN (вход
    # пользователя в систему)
    if data['type'] == 'LOGIN':
        notify_user_login(data)


def parse_input_line(parser: auparse.AuParser):
    """
    Функция, которая разбирает загруженную в парсер строку на отдельные события
    и записи, а затем вызывает обработчик для каждой записи
    """
    if not parser.first_record():
        # на вход была получена пустая строка или строка, не содержащая
        # информацию о событиях безопасности - выйти из функции
        return
    while True:
        while True:
            # вызывать функцию parse_record для каждой отдельной записи
            parse_record(parser)
            # метод parser.next_record "передвигает курсор" на следующую запись
            # события, если записей нет, будет возвращено значение False - в
            # таком случае необходимо переходить к обработке следующего события
            if not parser.next_record():
                break
        # метод parser.parse_next_event "передвигает курсор" на следующее
        # событие, если событий нет, будет возвращено значение False - в таком
        # случае необходимо завершить обработку текущей строки
        if not parser.parse_next_event():
            break


def main():
    """
    Функция, которая является точкой входа в программу.
    """
    # настройка обработчиков сигналов HUP (перечитать конфигурационный файл) и
    # TERM (завершить работу программы)
    hup_flag = False
    term_flag = False

    def sighup_handler(signal, frame):
        nonlocal hup_flag
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
        hup_flag = True

    def sigterm_handler(signal, frame):
        nonlocal term_flag
        term_flag = True

    signal.signal(signal.SIGHUP, sighup_handler)
    signal.signal(signal.SIGTERM, sigterm_handler)

    # войти в бесконечный цикл обработки событий
    while True:
        if hup_flag:
            # вызвать функцию для обновления настроек программы если получен
            # сигнал HUP
            reload_config()
            hup_flag = False
        elif term_flag:
            # завершить работу программы если получен сигнал TERM
            sys.exit(0)
        else:
            # считать строку, содержащую информацию об одном или нескольких
            # событиях аудита, из потока стандартного ввода stdin
            for line in sys.stdin:
                # инициализировать парсер записей журнала аудита AuParser,
                # поставляемый с библиотекой auparse.
                parser = auparse.AuParser(auparse.AUSOURCE_BUFFER, line)
                # вызвать функцию для разбора строки с информацией о событии
                parse_input_line(parser)


if __name__ == '__main__':
    sys.exit(main())
```

Чтобы служба аудита могла запустить расширение, файл с исполняемым кодом необходимо разместить в каталоге, доступном для чтения службой и сделать этот файл исполняемым. Обычно для этих целей используется каталог `/usr/local/bin`.

В нашем примере код на языке программирования Python не требует компиляции, так что достаточно будет просто сохранить его в файл `/usr/local/bin/audit-plugin-notify`, сделать его исполняемым и установить безопасные права:

```
$ sudo chown root:root /usr/local/bin/audit-plugin-notify
$ sudo chmod 755 /usr/local/bin/audit-plugin-notify
```

Для тестирования и/или отладки расширения нет необходимости сразу подключать его к службе аудита. Поскольку расширение работает с данными в том же формате, в котором они записываются в системный журнал событий безопасности, вы можете использовать для отладки вывод команды `ausearch --raw`:

```
$ sudo ausearch -m LOGIN --raw  | /usr/local/bin/audit-plugin-notify
```

Либо перенаправлять последние записи из журнала непосредственно на ввод расширения с помощью команды `tail -F`:

```
$ sudo tail -F /var/log/audit/audit.log | /usr/local/bin/audit-plugin-notify
```

Для отправки электронных писем из плагина вам понадобится локальная почтовая служба, например, `postfix`:

```
$ sudo dnf install -y postfix
$ sudo systemctl enable --now postfix
```

Для целей тестирования дополнительная конфигурация почтового сервиса не требуется — достаточно чтобы он был запущен и пользователь, на имя которого будет отправляться электронная почта, существовал (не забудьте внести соответствующие изменения в значение переменной `ADMIN_EMAILS` в исходном коде расширения). Полученная почта будет сохраняться в текстовый файл очереди `/var/spool/mail/ИМЯ_ПОЛЬЗОВАТЕЛЯ`, с которым может работать как почтовый клиент Evolution (тип сервера «Стандартная для Unix очередь типа mbox»), так и любая программа для просмотра текстовых файлов (`less`, `cat` и т.п.). Для получения информации о безопасной настройке почтовой службы для реальных условий вам необходимо обратиться к специализированному руководству.

После того как разработка расширения завершена, его можно подключать к службе аудита. Для этого необходимо создать соответствующий конфигурационный файл плагина в каталоге `/etc/audit/plugins.d`, имя файла может быть любым, но файл должен иметь расширение `.conf`. В нашем примере создадим файл `/etc/audit/plugins.d/audit-plugin-notify.conf` следующего содержания:

```
# включает (yes) или выключает расширение (no)
active = yes

# направление, в котором работает расширение. В настоящий момент значение всегда
# должно быть out.
direction = out

# путь к исполняемому файлу расширения.
path = /usr/local/bin/audit-plugin-notify

# для пользовательских расширений, не входящих в поставку службы аудита,
# значение type всегда должно быть always.
type = always

# опционально, вы можете передать через службу аудита до двух аргументов
# командной строки для запуска расширения. Например, путь к файлу или
# идентификатор пользователя. В нашем примере дополнительные аргументы не
# требуются.
# args = 1004

# служба аудита может подавать данные на ввод расширения в двух форматах:
# в своём внутреннем бинарном (binary) и в текстовом (string). Используемый
# нами модуль auparse поддерживает только текстовый формат.
format = string
```

Полная информация по всем опциям конфигурационного файла расширения доступна на странице руководства `man auditd-plugins`.

После создания файла установите для него корректные права:

```
$ sudo chown root:root /etc/audit/plugins.d/audit-plugin-notify.conf
$ sudo chmod 644 /etc/audit/plugins.d/audit-plugin-notify.conf
```

Для того чтобы расширение, запущенное службой аудита, могло отправлять почту, потребуется добавить соответствующее разрешение в политики SELinux. Создайте файл `audit-plugin-notify.te` следующего содержания:

```
module audit-plugin-notify 1.0;

require {
        type smtp_port_t;
        type sysctl_net_t;
        type auditd_t;
        class tcp_socket name_connect;
        class dir search;
        class file getattr;
        class file read;
        class file open;
}

#============= auditd_t ==============
allow auditd_t smtp_port_t:tcp_socket name_connect;
allow auditd_t sysctl_net_t:dir search;
allow auditd_t sysctl_net_t:file { getattr open read };
```

Затем скомпилируйте его и создайте SELinux модуль с политикой:

```
$ checkmodule -M -m -o audit-plugin-notify.mod audit-plugin-notify.te
$ semodule_package -o audit-plugin-notify.pp -m audit-plugin-notify.mod
```

После этого, используйте следующую команду для загрузки SELinux модуля:

```
$ sudo semodule -i audit-plugin-notify.pp
```

Подготовка к запуску расширения службы аудита завершена. Теперь достаточно подать службе сигнал `HUP` чтобы она перечитала свои конфигурационные файлы:

```
$ sudo auditctl --signal reload
```

В случае успешного запуска вы увидите, что служба аудита запустила плагин в дереве процессов:

```
$ ps axf | grep audit
     45 ?        S      0:00  \_ [kauditd]
    826 ?        S<sl   0:00 /sbin/auditd
  10342 ?        S<     0:00  \_ /usr/bin/python3 -I /usr/local/bin/audit-plugin-notify
```

Если процесс не запустился, то для диагностики вам необходимо будет просмотреть последние записи в журнале сервиса `auditd`:

```
$ sudo journalctl -u auditd
```

Теперь, когда расширение запущено, вы можете войти в систему от имени какого-либо пользователя и убедиться, что вы получили уведомление на почтовый адрес, указанный в переменной `ADMIN_EMAILS`.

Используя механизм расширений, вы можете реализовать любую логику обработки событий безопасности в реальном времени: автоматизировать действия, являющиеся реакцией на определённые события безопасности, отправлять уведомления через корпоративные каналы связи, реализовать интеграцию с системами, которые изначально не поддерживают данные в формате службы `auditd` и т.д.

SIEM-системы

SIEM (Security information and event management) система — это комплексная система управления информационной безопасностью. Как правило, такое программное обеспечение имеет клиент-серверную архитектуру и предназначено для централизованного решения широкого круза задач:

- сбор данных о безопасности из различных источников в компьютерной сети: серверов и рабочих станций, сетевых устройств и различного программного обеспечения;
- организация эффективного хранения полученных данных для последующей обработки;
- анализ событий безопасности, выявление потенциальных угроз и связей между различными событиями;
- автоматическая либо автоматизированная реакция на инциденты, связанные с безопасностью: отправка уведомлений, блокировка доступа и т.п.;
- аудит и отчётность на основе собранных данных.

С точки зрения службы `auditd` интеграция с SIEM-системами обычно реализуется одним из следующих способов:

- настройка подсистемы `auditd` на отправку журналов безопасности в SIEM-систему;
- SIEM-система предоставляет расширение для службы `auditd`, реализующее передачу данных;
- на клиентскую машину устанавливается специальный агент SIEM-системы, который отслеживает появление событий безопасности в системном журнале и отправляет их в SIEM-систему самостоятельно.

Операционная система МСВСфера может выполнять действия, являющиеся реакцией на события безопасности с применением сертифицированных средств защиты информации класса SIEM и систем обнаружения вторжений.

Типы записей подсистемы аудита

В таблице представлены некоторые типы записей, генерируемые подсистемой аудита в МСВСфера ОС.

Таблица 25: Типы записей, генерируемые подсистемой аудита в МСВСфера ОС

Тип записи	*Источник*	*Описание*
`ACCT_LOCK`	`user`	Пользовательская учётная запись была заблокирована администратором.
`ADD_GROUP`	`user`	Добавлена новая пользовательская группа.
`ADD_USER`	`user`	Добавлена новая пользовательская учётная запись.
`ANOM_ABEND`	`kernel`	Процесс завершился аварийно (segmentation fault и т.п.).
`AVC`	`kernel`	Отказ или предоставление разрешений SELinux AVC (Access Vector Cache).
`BPF`	`kernel`	Загрузка или выгрузка BPF (Berkeley Packet Filter).
`CONFIG_CHANGE`	`user`	Конфигурация подсистемы аудита была изменена.

продолжение на следующей странице

Таблица 25 – продолжение с предыдущей страницы

Тип записи	*Источник*	*Описание*
CRED_ACQ	user	Пользовательские учётные данные загружены в пользовательское пространство. См. описание функции pam_setcred модуля PAM.
CRED_DISP	user	Пользовательские учётные данные выгружены из пользовательского пространства. См. описание функции pam_setcred модуля PAM.
CRED_REFR	user	Пользовательские учётные данные были обновлены в пользовательском пространстве. См. описание функции pam_setcred модуля PAM.
CRYPTO_KEY_USER	user	Криптографический ключ был использован в криптографических целях.
CRYPTO_SESSION	user	Содержит параметры, использованные во время установления TLS-сессии.
CWD	kernel	Запись о текущем рабочем каталоге, из которого был запущен процесс, выполнивший системный вызов.
DAEMON_CONFIG	user	Конфигурация службы (сервиса) аудита была изменена.
DAEMON_START	user	Служба (сервис) аудита была запущена.
DEL_GROUP	user	Пользовательская группа была удалена.
DEL_USER	user	Пользовательская учётная запись удалена.
EXECVE	kernel	Содержит команду запуска процесса, присутствует только в событиях, связанных с системным вызовом execve(2).
GRP_MGMT	user	Изменены атрибуты пользовательской группы.
KERN_MODULE	kernel	Модуль ядра был загружен или выгружен.
LOGIN	user	Содержит информацию о входе пользователя в систему.
MAC_CONFIG_CHANGE	user	Был изменён логический переключатель SELinux (SELinux boolean).
PATH	kernel	Информация о пути, который был передан системному вызову в качестве аргумента.
PROCTITLE	kernel	Содержит полную команду запуска процесса, вызвавшего данное событие безопасности.
SERVICE_START	user	Запущена служба (сервис).
SERVICE_STOP	user	Остановлена служба (сервис).
SOCKADDR	kernel	Содержит информацию о сокете, присутствует только в событиях, связанных с этим сокетом.
SOFTWARE_UPDATE	user	Информация об обновлении программного обеспечения.
SYSCALL	kernel	Информация о выполненном системном вызове.
SYSTEM_BOOT	user	Система была загружена.
SYSTEM_RUNLEVEL	user	Изменение уровня выполнения системы (например, через telinit).
SYSTEM_SHUTDOWN	user	Система была остановлена.
USER_ACCT	user	Обнаружена попытка авторизации пользователя.
USER_AUTH	user	Обнаружена попытка аутентификации пользователя.
USER_CHAUTHTOK	user	Пароль пользователя был изменён.
USER_CMD	user	Команда была запущена из пользовательского пространства. В конфигурации по умолчанию протоколирует только запуск sudo.
USER_END	user	Пользовательская сессия была завершена.
USER_ERR	user	Ошибка состояния учётной записи пользователя.
USER_LOGIN	user	Пользователь вошёл в систему.
USER_LOGOUT	user	Пользователь вышел из системы.
USER_MGMT	user	Изменение атрибутов пользовательской учётной записи.
USER_ROLE_CHANGE	user	SELinux-роль пользователя изменилась.
USER_START	user	Запущена пользовательская сессия.
VIRT_CONTROL	user	Изменено состояние виртуальной машины (запущена, остановлена и т.д.).
VIRT_MACHINE_ID	user	Виртуальной машине назначен контекст безопасности SELinux.
VIRT_RESOURCE	user	Виртуальной машине был назначен (выделен) какой-либо ресурс.

В столбце «Источник» применяются следующие сокращения:

- user — источником события является пользовательское пространство;
- kernel — источником события является пространство ядра.

Параметр	Описание
log_file	Полное имя файла, в котором будут храниться данные регистрации событий безопасности.
log_group	Группа, являющаяся владельцем файла регистрации.
log_format	Формат хранения данных регистрации. Возможные значения: raw (данные записываются в том виде, в каком они были получены от ядра операционной системы) и nolog (запись данных отключается).
priority_boost	Приоритет выполнения службы регистрации.
flush	Режим работы службы регистрации. Возможные значения: none (не использовать какие-либо политики записи, т.е. дополнительные действия), incremental (запись с периодичностью, определенной параметром freq), data (запись данных в синхронном режиме), sync (запись в синхронном режиме и данных, и метаданных файла).
freq	Максимальное число регистрационных записей, которые могут храниться в буфере перед записью буферизованных данных на диск. Используется, только когда параметр flush имеет значение incremental.
num_logs	Максимальное число файлов регистрации на диске. Используется, только когда параметр max_log_file_action имеет значение rotate. Значение параметра не должно превышать 99.

продолжение на следующей странице

Таблица 26 – продолжение с предыдущей страницы

Параметр	Описание
disp_qos	Режим передачи данных между службой регистрации и диспетчером. Возможные значения: lossy (блокирование запрещено, т.е. служба регистрации может не передавать диспетчеру некоторые данные о событиях, если очередь данных о событиях полна. При этом данные регистрации будут записаны на диск, если только значение параметра log_format не равно nolog), lossless (блокирование разрешено, т.е. запись данных регистрации о событиях на диск будет остановлена, пока не освободится место в очереди).
dispatcher	Место расположения исполняемого файла программы диспетчера.
name_format	Порядок разрешения имен хостов. Возможные значения: none (имя не используется), hostname (имя, возвращенное через запрос gethostname), fqd (полное имя хоста, возвращенное через DNS запрос) numeric (IP-адрес), user (строка, определенная в параметре name).
max_log_file	Максимальный размер файла регистрации в мегабайтах, по достижении которого будет выполнено действие, определенное параметром max_log_file_action. Возможные действия: ignore (ничего не делать), syslog (отправить предупреждение в syslog), suspend (остановить запись данных регистрации событий на диск), rotate (произвести ротацию файлов регистрации в соответствии с параметром num_logs), keep_logs (осуществить ротацию, не удаляя при этом старые файлы).
space_left	Величина в мегабайтах, определяющая размер оставшегося дискового пространства, по достижении которого будет выполнено действие, определенное параметром space_left_action. Возможные действия: ignore (ничего не делать), syslog (отправить предупреждение в syslog), email (отправить письмо аккаунту, определенному в action_mail_acct), exec (выполнить скрипт), suspend (остановить запись на диск и перевести систему в single mode), halt (выключить систему).
admin_space_left	Величина в мегабайтах оставшегося свободного пространства на диске для предупреждения администратора о том, что надо добавить/очистить свободное пространство. Величина должна быть меньше чем space_left. Действия, которые можно определить в admin_space_left_action, аналогичны space_left_action.
disk_full_action	Действия, выполняемые при заполнении всего дискового пространства. Аналогичны space_left_action.
disk_error_action	Действия, выполняемые при возникновении дисковой ошибки. Аналогичны space_left_action.

Ограничение программной среды

Введение

Средства ограничения программной среды предоставляют возможности установки программного обеспечения доверенным образом; применения типовых наборов различных программных конфигураций; управления запуском программного обеспечения, в том числе определения запускаемых программ, настройки параметров запуска и контроля за их запуском; реагирования на попытки запуска, произведенные в нарушение установленных правил, а также другие возможности.

Включение программ в автозагрузку

Утилита `chkconfig` позволяет включать программы в автозагрузку с целью их автоматического запуска при старте операционной системы.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 27: Опции утилиты `chkconfig` и их значения

Опция	Значение
`--level levels`	Определяет уровни, на которых соответствующая программа должна выполняться. Уровни указываются на месте параметра `levels` в качестве строки целочисленных значений в диапазоне от 0 до 6. Так, например, при передаче опции `--level 35` утилите будет передано указание на уровни 3 и 5 соответственно.
`--no-redirect`	Если утилита запущена в системе, использующей утилиту `systemd` в качестве системы инициализации, то `chkconfig` будет перенаправлять команды в `systemd`, если у данной службы существует соответствующий файл, предназначенный для таких обращений. Данная опция отключает процесс перенаправления утилите `systemd` и обеспечивает работу только с символьными ссылками в директориях `/etc/rc[0-6].d`.
`--add name`	Добавляет новую службу для управления утилитой `chkconfig`. Имя службы указывается на месте параметра `name`.
`--del name`	Удаляет службу, имя которой указывается на месте параметра `name`, из-под управления утилитой `chkconfig`. Также из директорий `/etc/rc[0-6].d` удаляются любые символьные ссылки, указывающие на удаляемую службу.
`--override name`	Производит переопределение настроек службы, имя которой указывается на месте параметра `name`, вместо базовых настроек.
`--list name`	Выводит все службы, доступные для `chkconfig`, а также показывает их статус на каждом уровне (вкл/выкл). Если опции передать аргументом имя некоторой службы, которое указывается на месте параметра `name`, то будет выведена информация только об указанной службе.

Управление системными службами

Утилита `systemctl` позволяет управлять системными службами.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 28: Опции утилиты `systemctl` и их значения

Опция	Значение
`-t, --type`	Указывает на тип так называемого юнита (службы, сокета, устройства и т.п.). Может быть в виде списка наименований типов, разделенных запятой, если требуется указать более, чем на один тип.
`-a, --all`	При выведении списка юнитов вывести абсолютно все загруженные юниты вне зависимости от их статуса, включая те из них, которые являются неактивными.
`start [имя сервиса]`	Запускает работу сервиса с указанным именем.
`stop [имя сервиса]`	Останавливает работу сервиса с указанным именем.
`reload [имя сервиса]`	Перезагружает конфигурацию сервиса с указанным именем.
`restart [имя сервиса]`	Перезапускает сервис с указанным именем.
`try-restart [имя сервиса]`	Перезапускает сервис с указанным именем, если данный сервис уже работает на момент запуска утилиты.

продолжение на следующей странице

Таблица 28 – продолжение с предыдущей страницы

Опция	Значение
`reload-or-restart [имя сервиса]`	Перезагрузить конфигурацию сервиса с указанным именем, если сервис поддерживает такую команду, или выполнить перезапуск службы. Если на момент запуска утилиты указанная служба не была запущена, то она запустится после успешного выполнения команды.
`reload-or-try-restart [имя сервиса]`	Перезагрузить конфигурацию сервиса с указанным именем, если сервис поддерживает такую команду, или выполнить перезапуск службы. Если на момент запуска утилиты указанная служба не была запущена, то указанная команда не произведет никаких действий.
`kill [имя сервиса]`	Осуществить принудительную остановку работы службы с указанным именем.
`is-active [имя сервиса]`	Осуществляет проверку, активна ли на момент запуска утилиты служба с указанным именем. Если служба активна, или хотя бы одна из служб, переданных в качестве аргумента данной команде, активна (в случае, если были переданы наименования более, чем одной службы), выведется нулевое значение. В противном случае — ненулевое.
`is-failed [имя сервиса]`	Проверяет, были ли проблемы при запуске указанной службы или служб. Если хотя бы у одной из служб возникали проблемы, будет выведено нулевое значение.
`enable [имя сервиса]`	Добавляет указанный сервис (или их множество) в автозапуск.
`disable [имя сервиса]`	Убирает указанный сервис (или их множество) из автозапуска.
`is-enabled [имя сервиса]`	Проверяет, находится ли указанная служба (или службы, в случае, если в качестве аргумента был передан список наименований) в автозапуске. Если хотя бы одна из указанных служб находится в автозапуске, будет выведено нулевое значение.
`--version`	Вывести информацию о версии утилиты.
`-h, --help`	Вывести справочную информацию об утилите.

Пример: проверим статус сервера печати.

Для этого выполним следующую команду:

```
$ sudo systemctl status cups
cups.service - CUPS Printing Service
  Loaded: loaded (/usr/lib/systemd/system/cups.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
```

Пример: разрешим автоматический запуск сервера печати CUPS при загрузке системы.

Для этого выполним следующую команду:

```
$ sudo systemctl enable cups
Created symlink from /etc/systemd/system/multi-user.target.wants/cups.service to /usr/lib/
↪systemd/system/cups.service.
Created symlink from /etc/systemd/system/printer.target.wants/cups.service to /usr/lib/systemd/
↪system/cups.service.
Created symlink from /etc/systemd/system/sockets.target.wants/cups.service to /usr/lib/systemd/
↪system/cups.socket.
Created symlink from /etc/systemd/system/multi-user.target.wants/cups.path to /usr/lib/systemd/
↪system/cups.path.
```

Настройка запуска программ по расписанию

Утилита `crontab` позволяет настраивать запуск программ по расписанию.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 29: Опции утилиты `crontab` и их значения

Опция	Значение
`-u`	Указывает пользователя, чье расписание должно редактироваться.
`-l`	Вывод текущего файла расписания.
`-r`	Удаление текущего файла расписания.
`-e`	Редактирование файла расписания.

продолжение на следующей странице

Таблица 29 – продолжение с предыдущей страницы

Опция	Значение

Таблица расписания состоит из шести колонок, разделяемых пробелами или символами табуляции. Первые пять колонок задают время выполнения (минута, час, день, месяц, день недели). В них может находиться число, список чисел, разделённых запятыми, диапазон чисел, разделённых дефисом, символы `*` или `/`. После полей времени указывается пользователь, от которого запускается программа. Все остальные символы в строке интерпретируются как выполняемая программа с её параметрами.

Пример: установим с помощью утилиты `crontab` ограничения на доступ к системе по времени, с 10:28 до 10:30. Команда `passwd -l user2` блокирует возможность авторизации, дописывая символ восклицательного знака к строке пароля в файле `/etc/shadow`. Команда `passwd -u user2` производит обратную операцию, снимая тем самым блокировку. Заполним файл расписания и выполним команду `service crond restart`:

```
$ sudo crontab -e
crontab: Installing new crontab

$ sudo service crond restart
Redirecting to /bin/systemctl restart crond.service

$ sudo crontab -l
28 10 * * * /usr/bin/passwd -l user2
30 10 * * * /usr/bin/passwd -u user2
```

Управление программными пакетами

Утилита `rpm` позволяет управлять так называемыми программными пакетами, т.е. управлять их установкой, обновлением, проверкой и удалением.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 30: Опции утилиты `rpm` и их значения

Опция	Значение
`-i, --install`	Установка нового пакета.
`-u, --upgrade`	Установка или обновление уже установленного пакета до новой версии. При этом после установки пакета все другие версии удаляются.
`-f, --freshen`	Обновление пакета, но только если предыдущая версия уже установлена.
`--nodeps`	Не выполнять проверку зависимостей перед установкой или обновлением пакета.
`--nosuggest`	Не предлагать пакет(ы) для разрешения отсутствующих зависимостей.
`--noorder`	Не выполнять переупорядочивание пакетов для установки. Список пакетов обычно переупорядочивается для удовлетворения зависимостей.
`--oldpackage`	Разрешает обновить или заменить пакет более старой версией.
`--replacefiles`	Установить пакеты, даже если они заменяют файлы от других установленных пакетов.
`--replacepkgs`	Установить пакеты, даже если они уже установлены в систему.
`--includedocs`	Устанавливать файлы с документацией.
`--excludedocs`	Не устанавливать файлы с документацией.
`-e, --erase`	Удалить заданный пакет.
`--allmatches`	Удалить все версии пакета.
`--nodeps`	Не проверять зависимости перед удалением пакетов.
`--test`	Выполнить только проверку установки пакета.

продолжение на следующей странице

Таблица 30 – продолжение с предыдущей страницы

Опция	Значение
`-q, --query`	Вывести информацию о пакете.
`-a, --all`	Выполняет запрос ко всем установленным пакетам.
`--changelog`	Вывести информацию об изменениях в пакете.
`-l, --list`	Вывести список файлов в пакете.
`-P, --provides`	Вывести функциональность, предоставляемую пакетом.
`-R, --requires`	Вывести пакеты, от которых зависит этот пакет.
`-v, --verify`	Выполнить проверку метаданных пакета и его контрольной суммы.
`--version`	Вывести номер версии утилиты.
`--help`	Вывести справку об использовании утилиты.

Установка последней версии пакета/группы пакетов

Утилита `dnf` используется для установки последней версии пакета или группы пакетов с учетом существующих зависимостей.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 31: Опции утилиты `dnf` и их значения

Опция	Значение
`install`	Используется для установки последней версии пакета с учетом существующих зависимостей.
`reinstall`	Используется для переустановки пакета с идентичной версией.
`update`	Используется для обновления всех пакетов в системе.
`download`	Используется для загрузки пакета из репозитория.
`downgrade`	Используется для понижения версии пакета с версии, установленной на данный момент, до предыдущей самой высокой версии или указанной версии.
`remove`	Используется для удаления указанных пакетов из системы, а также для удаления пакетов, зависящих от удаляемых пакетов.
`info`	Используется для вывода описаний и общей информации о доступных пакетах.
`search`	Используется для поиска пакетов.
`list`	Используется для вывода различной информации о доступных пакетах.
`repolist all`	Используется для вывода списка всех репозиториев.
`clean`	Используется для удаления различных данных, накапливающихся со временем в кэше утилиты.
`history`	Используется для вывода истории использования утилиты.
`groupinstall`	Используется для установки последней версии всех пакетов из группы с учетом существующих зависимостей.
`groupupdate`	Используется для обновления всех пакетов из группы.
`groupremove`	Используется для удаления всех пакетов из группы.
`groupinfo`	Используется для вывода списка пакетов, относящихся к группе.
`grouplist`	Используется для вывода имен всех существующих групп пакетов.
`provides`	Используется, чтобы выяснить, какой пакет предоставляет тот или иной файл.
`repoquery --requires`	Вывести зависимости неустановленного пакета.
`repoquery --requires --resolve`	Вывести список пакетов, которые необходимы для удовлетворения зависимостей.
`-v, --verbose`	Запустить с большим количеством отладочной информации.
`-d, --debuglevel`	Устанавливает уровень отладки.
`-h, --help`	Вывести справку и выйти.

Стирание данных

Введение

Средства стирания данных предоставляют возможности безвозвратного удаления ставших ненужными данных и обеспечения недоступности остаточной информации путем многократной перезаписи использованных мест памяти специальными последовательностям.

Заполнение случайными числами места, занятого файлами

Утилита `shred` позволяет заполнять случайными числами место, занятое файлами.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 32: Опции утилиты `shred` и их значения

Опция	Значение
`-f, --force`	Изменить права для разрешения записи, если это необходимо.
`-n, --iterations=N`	Перезаписать файл N раз вместо 3-х по умолчанию.
`--random-source=FILE`	Перезаписать файл случайными данными, взятыми из файла с именем `FILE`.
`-s, --size=N`	Перезаписать только N байт. Можно использовать суффиксы `K`, `M`, `G` для указания размерности: килобайт, мегабайт, гигабайт.
`-u, --remove`	Обрезать и удалить файл после перезаписи. По умолчанию файлы не удаляются.
`-v, --verbose`	Показывать ход выполнения.
`-x, --exact`	Не округлять размер файла до следующего целого блока.
`-z, --zero`	На последней итерации перезаписать файл нулями.
`--version`	Показать версию утилиты и выйти.
`--help`	Показать справку и выйти.

Пример: заполним место, занятое файлом `filename`, с последующим удалением файла.

Для этого выполним следующую команду:

```
$ sudo shred -u -z filename
```

Стирание данных в свободном пространстве раздела, в котором находится директория

Утилита `sfill` позволяет стирать данные в свободном пространстве раздела, в котором находится заданная директория. Стирание производится в четыре шага:

1. Однократная перезапись числами 255 (0xFF).
2. Пятикратная перезапись случайными числами.
3. Двадцатисемикратная перезапись специальными числами.
4. И еще один раз пятикратная перезапись случайными числами.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 33: Опции утилиты `sfill` и их значения

Опция	Значение
`-f`	Выполнение более быстрым образом за счет пропуска второго и четвертого шагов перезаписи случайными числами.
`-i`	Очистка свободного пространства только индексного дескриптора, но не свободного пространства жесткого диска.
`-I`	Очистка свободного пространства только жесткого диска без затрагивания свободного пространства индексного дескриптора.
`-l`	Выполнение более быстрым образом за счет пропуска третьего и четвертого шагов или путем выполнения только одного шага перезаписи данных нулевыми значениями, если эту опцию задать дважды (например, `sdmem -l -l`).
`-v`	Работа будет сопровождаться выводом динамической строки, показывающей прогресс её выполнения.
`-z`	На четвертом шаге вместо перезаписи случайными числами выполнять перезапись нулями.

Пример: выполним очистку свободного пространства.

Для этого выполним следующую команду:

```
$ sudo sfill -vz /mnt/
Using /dev/urandom for random input.
Wipe mode is secure (38 special passes)
Wiping now ...
Creating /mnt/00000000.000 ... ************************** Wiping inodes ...
Done ... Finished
```

Стирание данных в разделах подкачки

Утилита `sswap` позволяет стирать данные в разделах подкачки. Алгоритм стирания данных абсолютно такой же, как и у утилиты `sfill`.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 34: Опции утилиты `sswap` и их значения

Опция	Значение
`-f`	Выполнение более быстрым образом за счет пропуска второго и четвертого шагов перезаписи случайными числами.
`-l`	Выполнение более быстрым образом за счет пропуска третьего и четвертого шагов или путем выполнения только одного шага перезаписи данных нулевыми значениями, если эту опцию задать дважды.
`-v`	Работа будет сопровождаться выводом динамической строки, показывающей прогресс её выполнения.
`-z`	На четвертом шаге вместо перезаписи случайными числами выполнять перезапись нулями.

Стирание данных в оперативной памяти

Утилита `sdmem` позволяет стирать данные в оперативной памяти. Алгоритм стирания данных почти такой же, как и у утилиты `sfill`, но с тем отличием, что на первом шаге однократная перезапись производится числами 0 (0x00).

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 35: Опции утилиты `sswap` и их значения

Опция	Значение
`-f`	Выполнение более быстрым образом за счет пропуска второго и четвертого шагов перезаписи случайными числами.
`-l`	Выполнение более быстрым образом за счет пропуска третьего и четвертого шагов или путем выполнения только одного шага перезаписи данных нулевыми значениями, если эту опцию задать дважды.
`-v`	Работа будет сопровождаться выводом динамической строки, показывающей прогресс её выполнения.

Контроль целостности

Введение

Средства контроля целостности предоставляют возможности контроля целостности обрабатываемых данных и используемого программного обеспечения.

Вычисление и сверка контрольной суммы файла

Утилита `sha256sum` позволяет вычислять контрольные суммы файлов по алгоритму SHA-256 и осуществлять их сверку с другими контрольными суммами, хранящимися в файле.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 36: Опции утилиты `sha256sum` и их значения

Опция	Значение
`-b, --binary`	Позволяет считывать данные из файлов в двоичном режиме.
`-c, --check`	Позволяет осуществить сверку рассчитанного значения контрольной суммы с некоторым другим значением контрольной суммы, хранящимся в файле, имя которого должно быть передано утилите в качестве аргумента.
`--tag`	Выводить рассчитанную контрольную сумму в формате BSD.
`-t, --text`	Читать в текстовом режиме (по умолчанию).
`-z, --zero`	Завершать каждую выводимую строку NUL, а не символом новой строки и выключить экранирование имени файла.
`--ignore-missing`	Не сообщать или прерываться при отсутствии файлов.
`--quiet`	При сверке контрольных сумм позволяет не выводить сообщение `OK` для каждого успешного случая сверки контрольных сумм.
`--status`	При сверке контрольных сумм позволяет в конце работы утилиты не выводить ничего, кроме кода сверки контрольных сумм.
`--strict`	При сверке контрольных сумм позволяет выводить ненулевое значение для неправильно отформатированных строк контрольной суммы.
`-w, warn`	При сверке контрольных сумм позволяет выводить предупреждения о неправильно отформатированных строках контрольной суммы.
`--version`	Показать версию утилиты и выйти.
`--help`	Показать справку и выйти.

Пример: подсчитаем контрольную сумму файла с журналом аудита по алгоритму SHA-256.

Для этого выполним следующую команду:

```
$ sudo sha256sum /var/log/audit/audit.log
65sh8467h74j8kf36mf76356r853k25748864ud7k835hr2kwuv9l4q73m74x7a49 /var/log/audit/audit.log
```

Также ОС МСВСфера предоставляет возможность вычислять контрольные суммы файлов по алгоритму ГОСТ. Для этого необходимо установить пакет `gostsum`.

Пример: подсчитаем контрольную сумму файла с журналом аудита по алгоритму ГОСТ.

Для этого выполним следующую команду:

```
$ sudo gostsum /var/log/audit/audit.log
054h7j3a8af6kf2h9lk257r67k2974rfh47a5hg34n3h7s25ak3674wrmc475ls5 /var/log/audit/audit.log
```

Проверка целостности данных

Утилита `aide` предоставляет возможности проверки целостности данных.

1. Перед началом использования необходимо установить утилиту `aide` с помощью следующей команды (необходимо по запросу системы указать пароль суперпользователя root):

```
$ sudo dnf install aide
```

2. Затем необходимо инициализировать базу данных для хранения состояний файлов с помощью следующей команды (необходимо по запросу системы указать пароль суперпользователя root):

```
$ sudo aide --init
```

В конфигурации по умолчанию команда `aide --init` проверяет только файлы и директории, заданные в конфигурационном файле `/etc/aide.conf`. Чтобы добавить необходимые файлы и директории в базу AIDE, а также изменить параметры, необходимо внести соответствующие изменения в конфигурационный файл `/etc/aide.conf`.

3. Перед началом использования инициализированной базы данных удалите `.new` из имени файла с помощью следующей команды:

```
$ sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

4. Для изменения места расположения базы AIDE необходимо в файле `/etc/aide.conf` изменить значение параметра `DBDIR`. Для обеспечения дополнительной безопасности рекомендуется хранить базу данных, конфигурацию и файл `/usr/sbin/aide` в безопасном месте.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице.

Таблица 37: Опции утилиты `aide` и их значения

Опция	Значение
`-i, --init`	Инициализирует базу данных хранения состояний файлов. Созданная после инициализации база данных должна быть размещена в таком месте, из которого при осуществлении последующей проверки данные могли бы быть считаны. В противном случае при попытке осуществления проверки будет выведена информация об ошибке считывания.
`-C, --check`	Проверяет базу данных состояний файлов на наличие несоответствий, для чего необходимо иметь инициализированную базу данных. Эта опция выполняется по умолчанию, если не указать никакой другой опции.
`-u, --update`	Проверяет базу данных и обновляет её. При этом входные и выходные базы данных должны быть разными.
`-E, --compare`	Сравнивает две базы данных, которые должны быть определены в конфигурационном файле.
`-D, --config-check`	Осуществляет считывание данных из конфигурационного файла, уведомляя обо всех обнаруженных ошибках.
`-c, --config=configfile`	Позволяет считывать конфигурацию из указанного файла вместо используемого по умолчанию конфигурационного файла `./aide.conf`.
`-B, --before="configparameters"`	Позволяет обрабатывать указанные конфигурационные параметры до считывания данных из конфигурационного файла.
`-A, --after="configparameters"`	Позволяет обрабатывать указанные конфигурационные параметры после считывания данных из конфигурационного файла.

продолжение на следующей странице

Таблица 37 – продолжение с предыдущей страницы

Опция	Значение
`-V verbosity_level, --verbose=verbosity_level`	Определяет детальность обработки данных. На месте параметра `verbosity_level` может находиться целочисленное значение в диапазоне от `0` до `255`. По умолчанию значение данного параметра равно `5`. Если вызвать данную опцию, но не присвоить целочисленное значение параметру `verbosity_level`, то ему автоматически будет присвоено значение, равное `20`. Этот параметр переопределяет значение, установленное в конфигурационном файле.
`-r reporter, --report=reporter`	Определяет место (URL), в которое утилита должна отправлять свои результаты работы.

Пример: проверим систему, сравнив `aide.db` и текущее состояние системы, без обновления `aide.db.new`.

Для этого выполним следующую команду:

```
$ sudo aide --check
AIDE, version 0.15.1
### All files match AIDE database. Looks okay!
```

Если произошло изменение файлов системы, то будет получено соответствующее уведомление:

```
$ sudo aide --check
AIDE 0.15.1 found differences between database and filesystem!
Start timestamp: 2023-03-28 09:12:21
Summary:
   Total number of files: 264416
   Added files:           1
   Removed files:         0
   Changed files:         3
```

Рекомендуется проводить проверку каждый день. Вы можете настроить проведение ежедневной проверки с помощью команды `cron`. Например, настроим проверку на 03:00, добавив в файл `/etc/crontab` следующую строку:

```
00 3 * * * root /usr/sbin/aide --check
```

Также, мы рекомендуем обновлять основную базу AIDE после обновлений системы и пакетов, а также изменений в конфигурационных файлах.

Для обновления основной базу AIDE выполните следующую команду:

```
$ sudo aide --update
```

Будет создан файл `/var/lib/aide/aide.db.new.gz`. Чтобы начать использовать вновь созданный файл для проверки целостности данных, удалите `.new` из имени файла.

Обеспечение надёжного функционирования

Введение

Средства обеспечения надёжного функционирования предоставляют возможности резервного копирования и восстановления данных и программного обеспечения при сбоях и отказах, а также возможности функционирования отдельных экземпляров системы на нескольких технических средствах в отказоустойчивом режиме, обеспечивающем доступность сервисов и данных при выходе из строя одного из технических средств или при исчерпании вычислительных ресурсов.

Архивация файлов и директорий

Утилита `tar` позволяет архивировать файлы и директории со всеми их поддиректориями и файлами, а затем восстанавливать их из архива, т.е. является удобным средством для создания резервных копий.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленным в таблице:

Таблица 38: Опции утилиты `tar` и их значения

Опция	Значение
`-c, --create`	Создать новый архив.
`-r, --append`	Присоединить файлы к концу архива.
`--delete`	Удалить файл из архива.
`-t, --list`	Вывести список содержимого архива.
`-A, --catenate, --concatenate`	Присоединить существующий архив к другому архиву.
`-x, --extract, -get`	Извлечь файлы из архива.
`-u, --update`	Добавить в архив более новые версии файлов.
`-C, --directory=DIR`	Сменить директорию перед выполнением операции на `DIR`.
`--f, --file=ARCHIVE`	Вывести результат в архивный файл или в устройство `ARCHIVE`.
`-d, --diff`	Осуществить проверку на наличие различий между архивом и некоторой файловой системой.
`-v, --verbose`	Выводить подробную информацию о процессе выполнения команды.

Пример: в примере директория `mydir` и все её поддиректории сначала сохраняются в файле `myarch.tar`:

```
$ tar cf myarch.tar mydir
```

а затем извлекаются из архива:

```
$ tar xf myarch.tar
```

А этот скрипт организует хранение четырех последних резервных копий директории `/var/www` в директории `/opt/backup/www-backup`. Первая версия будет всегда иметь номер 0, последняя — номер 3. При создании новых версий старые будут удаляться. Сами резервные копии хранятся в сжатом виде.

```
#! /bin/bash
cd /opt/backup/www-backup
rm www-dump-3.tar.gz
cp www-dump-2.tar.gz www-dump-3.tar.gz
cp www-dump-1.tar.gz www-dump-2.tar.gz
cp www-dump-0.tar.gz www-dump-1.tar.gz
tar --selinux --acls --xattrs --czf www-dump-o.tar.gz /var/www
```

Создание архивов и извлечение файлов из них

Утилита `cpio` используется для создания архивов и извлечения файлов из них, а также для копирования файлов в целях их переноса из текущей директории в другую. Поддерживает множество различных архивных форматов. При извлечении файлов из архива утилита автоматически распознает, каким типом обладает архив, с которым она взаимодействует.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 39: Опции утилиты `cpio` и их значения

Опция	Значение
`-o, --create`	Копировать файлы в архив.
`-A, --append`	Добавить файлы в архив. Может быть использована только в связке с опцией `-o`.
`-i, --extract`	Копирует файлы из архива или выводит список содержимого некоторого архива.
`-p, --pass-through`	Копирует файлы из одной файловой структуры в другую, комбинируя при этом режимы работы, использующиеся при передаче опций `-i` и `-o`, но не используя при этом архивы.
`-a, --reset-access-time`	Сбрасывает времена обращения к входным файлам после их копирования, так что при использовании данной опции будет нельзя распознать, что файлы были скопированы.

Пример: в примере сначала флеш-носитель монтируется как устройство `/mnt`:

```
$ mount /dev/sdb4 /mnt
```

Затем создается и записывается на флеш-носитель резервная копия директории `/lib`:

```
$ find /lib/ | cpio -o > /mnt/2/backup.cpi
```

Для того чтобы восстановить все файлы в директорию `/lib` из созданной ранее архивной копии, необходимо выполнить следующую команду:

```
$ cpio -ivmd /lib/\* < /mnt/2/backup.cpio
```

Резервное копирование данных

Утилита `amanda` обладает возможностью резервного копирования данных, хранящихся на множестве компьютеров в вычислительной сети. Она реализует клиент-серверную модель и использует следующие утилиты:

- **клиентская утилита `amandad`, взаимодействующая с сервером системы.**

  Во время своего выполнения вызывает другие утилиты:

  - `selfcheck` (проверка конфигурации клиента);
  - `sendsize` (оценка объема резервной копии);
  - `sendbackup` (выполнение операции резервного копирования);
  - `amcheck` (проверка конфигурации системы).

- **серверная утилита `amdump`, инициирующая все операции резервного копирования.**

  Во время своего выполнения использует другие утилиты и контролирует их выполнение:

  - `planner` (определение того, что надо копировать);
  - `driver` (интерфейс к внешнему устройству);
  - `dumper` (связывается с клиентским процессом `amandad`);
  - `taper` (запись данных на внешнее устройство);
  - `amreport` (подготовка сообщения о выполненном копировании).

- **административные утилиты:**

  - `amcheck` (проверка готовности системы к работе);
  - `amlabel` (записать метку на сменный носитель перед использованием в системе);
  - `amcleanup` (очистить систему после неплановой перезагрузки сервера или после непланового завершения операции резервного копирования);
  - `amflush` (переписать данные из дискового кэша на внешний носитель);
  - `amadmin` (выполнение большого количества различных административных операций).

- **утилиты восстановления данных:**

  - `amrestore` (восстановление данных с носителей, на которых записаны резервные копии, выполненные системой);
  - `amrecover` (программа для интерактивного восстановления данных с резервных копий).

Создание дисковых RAID-массивов

Утилита `mdadm` позволяет создавать так называемые дисковые RAID-массивы с использованием технологии распределения данных по нескольким дискам с целью достижения избыточности, отказоустойчивости, сокращения задержек и/или увеличения скорости чтения и записи, а также для улучшения возможностей восстановления данных в случае отказа.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 40: Опции утилиты `mdadm` и их значения

Опция	Значение
`-A, --assemble`	Режим сборки ранее созданного массива и его активации.
`-B, --build`	Режим сборки массива без суперблоков.
`-C, --creat`	Режим сборки нового массива.
`-F, --follow, --monitor`	Режим слежения за состоянием устройств.
`-G, --grow`	Режим расширения или уменьшения размера массива.
`-N, --name`	Устанавливает имя массива.
`-n, --raid-devices`	Указывает количество активных устройств в массиве.
`-x, --space-device`	Указывает количество запасных устройств в массиве.
`-z, --size`	Указывает объем пространства, используемого для каждого диска.
`-l, --level`	Устанавливает уровень массива.
`-c, --config`	Указывает файл конфигурации. По умолчанию `/etc/mdadm.conf`.
`-f, --fail`	Помечает перечисленные устройства как неисправные.
`-S, --stop`	Деактивирует массив и освобождает все ресурсы.
`-V --version`	Выводит информацию о версии утилиты.
`-h, --help`	Выводит справку об утилите.

Фильтрация сетевого потока

Введение

Средства фильтрации сетевого потока предоставляют возможности фильтрации входящих и исходящих сетевых потоков на основе установленного набора правил с учетом атрибутов безопасности и используемых сетевых протоколов, а также управления правилами фильтрации сетевых потоков; регистрации и учета выполнения проверок при фильтрации сетевых потоков.

Настройка файрвола (брандмауэра)

Утилита `firewall-cmd` позволяет настраивать работу файрвола (брандмауэра), осуществляющего фильтрацию сетевых потоков при помощи определения так называемых зон, иными словами, наборов правил, которые управляют трафиком на основе уровня доверия к той или иной сети.

Существуют следующие зоны:

- `drop` — самый низкий уровень доверия к сети. Весь входящий трафик сбрасывается без ответа. Поддерживаются только исходящие соединения;
- `block` — эта зона похожа на предыдущую, но при этом входящие запросы сбрасываются с сообщением `icmp-host-prohibited` или `icmp6-adm-prohibited`;
- `public` — эта зона представляет публичную сеть, которой нельзя доверять, однако поддерживает входящие соединения в индивидуальном порядке;
- `external` — зона внешних сетей. Поддерживает маскировку NAT, благодаря чему внутренняя сеть остается закрытой, но с возможностью получения доступа;
- `internal` — обратная сторона зоны `external`. Компьютерам в этой зоне можно доверять.

Доступны дополнительные сервисы:

- `dmz` — используется для компьютеров, расположенных в DMZ (зонах изолированных компьютеров, которые не будут иметь доступа к остальной части сети). Поддерживает только некоторые входящие соединения;
- `work` — зона рабочей сети. Большинству машин в сети можно доверять, доступны дополнительные сервисы;
- `home` — зона домашней сети. Окружению можно доверять, но поддерживаются только определённые пользователем входящие соединения;
- `trusted` — всем машинам в сети можно доверять.

Режимы работы утилиты и выполняемые функции задаются набором опций, перечисленных в таблице:

Таблица 41: Опции утилиты `firewall-cmd` и их значения

Опция	Значение
`--state`	Вывести состояние файрвола.
`--reload`	Перезагрузить правила из постоянной конфигурации.
`--complete-reload`	Жёсткая перезагрузка правил с разрывом всех соединений.
`--runtime-to-permanent`	Перенести настройки `runtime` в постоянную конфигурацию.
`--permanent`	Использовать постоянную конфигурацию.
`--get-default-zone`	Отобразить зону, используемую по умолчанию.
`--set-default-zone`	Установить зону по умолчанию.
`--get-active-zones`	Отобразить активные зоны.
`--get-zones`	Отобразить все доступные зоны.
`--get-services`	Вывести предопределенные сервисы.
`--list-all-zones`	Вывести конфигурацию всех зон.
`--new-zone`	Создать новую зону.
`--delete-zone`	Удалить зону.
`--list-all`	Вывести всё, что добавлено, из выбранной зоны.
`--list-services`	Вывести все сервисы, добавленные к зоне.
`--add-service`	Добавить сервис к зоне.
`--remove-service`	Удалить сервис из зон.
`--list-ports`	Отобразить порты, добавленные к зоне.
`--add-port`	Добавить порт к зоне.
`--remove-port`	Удалить порт из зоны.
`--query-port`	Показать, добавлен ли порт к зоне.
`--list-protocols`	Вывести протоколы, добавленные к зоне.
`--add-protocol`	Добавить протокол к зоне.
`--remove-protocol`	Удалить протокол из зоны.
`--list-source-ports`	Вывести порты источника, добавленные к зоне.
`--add-source-port`	Добавить порт-источник к зоне.
`--remove-source-port`	Удалить порт-источник из зоны.
`--list-icmp-blocks`	Вывести список блокировок icmp.
`--add-icmp-block`	Добавить блокировку icmp.
`--remove-icmp-block`	Удалить блокировку icmp.
`--add-forward-port`	Добавить порт для перенаправления в NAT.
`--remove-forward-port`	Удалить порт для перенаправления в NAT.
`--add-masquerade`	Включить NAT.
`--remove-masquerade`	Удалить NAT.

Пример: настройка правила блокировки адреса получателя может выглядеть следующим образом:

```
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0 -d 192.168.10.20 -j DROP
success
```

Пример: настройка правила отбрасывания всех входящих соединений по протоколу IPv4 может выглядеть следующим образом:

```
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT 0 -j DROP
success
```

Пример: настройка правила отбрасывания всех исходящих пакетов UDP может выглядеть следующим образом:

```
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0 -p upd -j DROP
success
```

Конфигурационный файл /etc/firewalld/firewalld.conf

Конфигурационный файл `/etc/firewalld/firewalld.conf` содержит основные параметры конфигурации для файрвола `firewalld`.

- `DefaultZone` — устанавливает зону по умолчанию для соединений или интерфейсов;

- `MinimalMark` — с этой опцией блок меток может быть зарезервирован для частного использования. Используются только отметки над этим значением. Значение по умолчанию равно `100`;
- `CleanupOnExit` — если `firewalld` останавливается, он очищает все правила. Если для этого параметра установлено значение `no` или `false`, текущие правила останутся нетронутыми. Значением по умолчанию является `yes` или `true`;
- `Lockdown` — если эта опция включена, изменения `firewalld` с интерфейсом D-Bus будут ограничены приложениями, которые перечислены в белом списке блокировки. Значением по умолчанию является `no` или `false`;
- `IPv6_rpfilter` — если эта опция включена, выполняется проверка фильтра обратного пути для пакета для IPv6. Если ответ на пакет будет отправлен через тот же интерфейс, на который поступил пакет, пакет совпадет и будет принят. В противном случае он будет отброшен. Для IPv4 `rp_filter` управляется с помощью `sysctl`;
- `IndividualCalls` — если этот параметр отключен, используются комбинированные вызовы restore, а не отдельные вызовы, чтобы применить изменения к файрволу. Использование отдельных вызовов увеличивает время, необходимое для применения изменений;
- `LogDenied` — добавление правил ведения журнала непосредственно перед отклонением и удалением правил в цепочках `INPUT`, `FORWARD` и `OUTPUT` для правил по умолчанию, а также окончательных правил отклонения и отбрасывания в зонах для настроенного типа пакета канального уровня. По умолчанию установлено `off` — отключение ведения журнала.
- `AutomaticHelpers` — для безопасного использования протокола IPv4 `iptables` и помощников по отслеживанию соединений этот параметр рекомендуется отключить. Возможные значения: `yes`, `no`, `system`. По умолчанию установлено `system`;
- `FirewallBackend` — выбирает реализацию брандмауэра. Возможные значения: `nftables` (по умолчанию) или `iptables`. Это относится ко всем примитивам `firewalld`. Единственным исключением являются прямые и сквозные правила, которые всегда используют традиционные `iptables`, `ip6tables` и `ebtables`.

Предупреждение

Внимание! Не следует выдавать права `sudo` на утилиту `iptables-save`. Утилита `iptables-save` при помощи ключа `-f` позволяет записывать дамп правил в файл. Соответственно, имея права `sudo`, пользователь сможет переписать любой файл в системе.

Создание виртуальной машины

Создание виртуальной машины с помощью утилиты virt-install

Одним из самых простых способов создания виртуальных машин с помощью командной строки является утилита `virt-install`.

Для вызова команды `virt-install` используется стандартный синтаксис:

```
$ virt-install [аргументы]
```

`virt-install` поддерживает как графический режим установки операционной системы с использованием протоколов VNC или SPICE, так и установку в текстовом режиме с помощью последовательной консоли. Во время создания виртуальной машины она может быть настроена на использование одного или нескольких дисков, сетевых интерфейсов, аудио устройств, аппаратных USB или PCI устройств и т.д.

В качестве установочного носителя может использоваться ISO-образ или виртуальный CD-ROM накопитель, установочное дерево дистрибутива, доступное по протоколам HTTP, HTTPS, FTP либо размещённое локально. Также поддерживается сетевая загрузка с использованием протокола PXE, импорт готовых образов дисков, полностью автоматическая установка операционной системы с помощью kickstart-файлов или опции `--unattended`.

Аргументы командной строки virt-install

У многих аргументов команды `virt-install` есть дополнительные параметры, которые указываются следующим образом: `--аргумент опция1=значение опция2=значение`. Используйте синтаксис `--аргумент=?` чтобы увидеть полный список таких параметров, например:

```
$ virt-install --disk=?
```

Большинство аргументов `virt-install` являются опциональными. В случае задания значения опции `--os-variant` либо успешного автоматического определения типа гостевой системы, для таких аргументов будут использованы соответствующие значения по умолчанию, определённые профилем устанавливаемой операционной системы. Профили предоставляются пакетом `osinfo-db`. В случае отсутствия профиля для устанавливаемой ОС потребуется определить как минимум следующие опции: `--memory`, настройки хранилища (`--disk` или `--filesystem`) и метод установки (`--cdrom`, `--location`).

При установке ОС МСВСфера через профили используются следующие значения:

- `jeos` — минимальная серверная конфигурация МСВСФера Сервер нужной версии.
- `desktop` — конфигурация для рабочих станций МСВСфера АРМ нужной версии.

Аргументы, передаваемые утилите `virt-install`, можно условно сгруппировать по их назначению:

- Параметры подключения — определяют тип используемого гипервизора и путь (ссылку) для подключения к нему.
- Общие параметры — общие параметры, применимые ко всем типам гостевых систем.
- Параметры установки — определяют каким образом будет выполняться установка гостевой операционной системы.
- Параметры гостевой системы — задают тип устанавливаемой операционной системы либо управляют настройками автоматического определения типа.
- Параметры хранилища — опции, связанные с настройкой хранилища виртуальной машины.
- Параметры сети — опции, связанные с настройкой сети виртуальной машины;
- Параметры графики — опции, связанные с настройкой графической подсистемы виртуальной машины.
- Параметры виртуализации — опции для переопределения используемого механизма виртуализации.
- Параметры устройств — опции для подключения физических и виртуальных устройств к виртуальной машине.
- Другие опции — опции, не вошедшие ни в одну из предыдущих групп.

Примеры использования virt-install

Следующая команда создаст в пользовательской сессии QEMU виртуальную машину `msvsphere-10-server` с двумя гигабайтами оперативной памяти, двумя виртуальными процессорами и виртуальным qcow2-диском объёмом двадцать гигабайт. Виртуальная машина будет запущена в режиме BIOS, в качестве установочного носителя будет использован ISO-образ `/srv/iso/MSVSphere-10.0-x86_64-server.iso`:

```
$ virt-install --name msvsphere-10-server \
      --cdrom /srv/iso/MSVSphere-10.0-x86_64-server.iso \
      --memory 2048 --vcpus 2 --disk size=20 --os-variant msvsphere10
```

Установка операционной системы МСВСфера 10 в режиме UEFI с отключённой поддержкой Secure Boot, в качестве источника установки используется установочное дерево дистрибутива, размещённое на официальном зеркале:

```
$ virt-install --name msvsphere-10-server \
      --memory 2048 --vcpus 2 --disk size=20 --os-variant msvsphere10 \
      --location https://repo1.msvsphere-os.ru/msvsphere/10/BaseOS/x86_64/os/ \
      --boot uefi,loader=/usr/share/edk2/ovmf/OVMF_CODE.fd, \
      loader_ro=yes,loader_type=pflash,nvram_template=/usr/share/edk2/ovmf/OVMF_VARS.fd,loader_
↪secure=no
```

Следующая команда создаст виртуальную машину в системной сессии QEMU и выполнит автоматическую установку операционной системы МСВСфера 10 в режиме

UEFI с включённой поддержкой Secure Boot, сценарий установки определён в kickstart-файле `msvsphere-10.ks`:

```
$ virt-install --name msvsphere-10-server --connect qemu:///system \
      --memory 2048 --vcpus 2 --disk size=20 --os-variant msvsphere10 \
      --location https://repo1.msvsphere-os.ru/msvsphere/10/BaseOS/x86_64/os/ \
      --boot uefi,loader=/usr/share/edk2/ovmf/OVMF_CODE.secboot.fd,loader_ro=yes, \
      loader_type=pflash,nvram_template=/usr/share/edk2/ovmf/OVMF_VARS.secboot.fd,loader_
↪secure=yes \
      --initrd-inject msvsphere-10.ks --extra-args "inst.ks=file:/msvsphere-10.ks"
```

Пример kickstart-файла (`msvsphere-10.ks` в примере выше) для автоматической установки системы в минимальной конфигурации без графического интерфейса:

```
# путь к установочному дереву дистрибутива
url --url https://repo1.msvsphere-os.ru/msvsphere/10/BaseOS/x86_64/kickstart/

# список репозиториев, которые необходимо подключить во время установки
repo --name=BaseOS --baseurl=https://repo1.msvsphere-os.ru/msvsphere/10/BaseOS/x86_64/os/
repo --name=AppStream --baseurl=https://repo1.msvsphere-os.ru/msvsphere/10/AppStream/x86_64/os/

# выполнять установку в текстовом режиме
text
# не выполнять настройку графического сервера Xorg/Wayland
skipx
# автоматически принимать условия лицензии
eula --agreed
# не запускать ассистента по настройке во время первого запуска
firstboot --disabled

# использовать английский язык как во время установки, так и на установленной
# системе. Дополнительно включить поддержку русского языка
lang en_US --addsupport=ru_RU
# настраивает раскладку клавиатуры, в данном случае будет использоваться только
# английская в американском варианте
keyboard us
# установить часовой пояс в московское время (GMT+3), флаг --utc указывает на
# то, что аппаратные часы хранят время в часовом поясе UTC
timezone Europe/Moscow --utc

# автоматически настроить сеть используя протокол DHCP
network --bootproto=dhcp
# включить брандмауэр и отрыть доступ по протоколу SSH
firewall --enabled --service=ssh
# отключить службу kdump и включить службы chronyd, rsyslog и sshd
services --disabled="kdump" --enabled="chronyd,rsyslog,sshd"
# включить SELinux
selinux --enforcing

# настроить вывод на последовательный порт, чтобы можно было подключаться к
# консоли виртуальной машины без графической сессии
bootloader --timeout=1 --append="console=tty0 console=ttyS0,115200n8 no_timer_check␣
↪crashkernel=auto net.ifnames=0"
# использовать только диск /dev/vda во время установки
ignoredisk --only-use=vda
# создать новую таблицу разделов на диске /dev/vda
clearpart --initlabel --drives=vda
# использовать автоматическую разбивку диска без отдельного раздела /home
autopart --nohome

# заблокировать вход пользователем root
rootpw --lock
# создать пользователя msvsphere с паролем msvsphere, сделать его
# администратором путём добавления в группу wheel
user --groups="wheel" --name msvsphere --password="msvsphere"

# автоматически перезагрузить систему после завершения установки и извлечь
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# установочный носитель
reboot --eject

# блок %packages определяет какие пакеты, группы и модули необходимо установить.
# опция --inst-langs определяет список языков, для которых необходимо добавлять
# поддержку
%packages --inst-langs=en,ru
# установить пакеты из группы core
@core
# установить пакеты из группы guest-agents
@guest-agents
# установить пакеты самоидентификации серверного варианта ОС МСВСфера
sphere-release-identity-server
sphere-release-server
sphere-release
%end

# отключить расширение kdump
%addon com_redhat_kdump --disable
%end
```

На системах с графическим интерфейсом после запуска команды `virt-install` автоматически запустится программа `virt-viewer`, с помощью которой вы сможете взаимодействовать с виртуальной машиной и выполнить установку операционной системы.

На системах без графического интерфейса вы можете использовать последовательную консоль, если устанавливаемая операционная система поддерживает такой режим. Для большинства ОС на базе GNU/Linux будет достаточно передать следующие аргументы команде `virt-install`:

```
$ virt-install ... -console pty,target_type=virtio --graphics none \
      --serial pty --extra-args 'console=ttyS0,115200n8'
```

Вы можете использовать рассмотренную схему для установки любых версий ОС МСВСфера.

Контейнеризация

Trivy

Trivy — это комплексный и универсальный сканер уязвимостей в образах контейнеров, файловых системах и репозиториях Git. **Trivy** имеет `сканеры`, которые ищут проблемы безопасности, и `цели`, где он может их найти.

Цели (которые может сканировать **Trivy**):

- Изображение контейнера
- Файловая система
- Репозиторий Git (удалённый)
- Образ виртуальной машины
- Kubernetes

Сканеры (то, что **Trivy** может там найти):

- Используемые пакеты ОС и программные зависимости (SBOM)
- Известные уязвимости (CVE)
- Проблемы и неправильные конфигурации IaC
- Конфиденциальная информация и секреты
- Лицензии на программное обеспечение

Trivy поддерживает большинство популярных языков программирования, операционных систем и платформ. Полный список см. на странице Scanning Coverage.

Установка

```
$ sudo dnf install -y trivy
```

Использование

```
trivy [глобальные флаги] команда [флаги] цель
trivy команда
```

Доступные команды

Команды сканирования

```
config             сканировать файлы конфигурации
filesystem  (fs)   сканировать локальную файловую систему
image       (i)    сканировать образ контейнера
kubernetes  (k8s)  [Экспериментально] сканировать кластер кубернетес
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
repository  (repo) сканировать git-репозиторий (удалённо)
rootfs             Сканировать rootfs
sbom               сканировать используемые пакеты ОС и программные зависимости (SBOM)
vm                 [Экспериментально] сканировать образ виртуальной машины
```

Команды управления

```
module      Управление модулями
plugin      Управление плагинами
vex         [Экспериментально] VEX утилиты
```

Вспомогательные команды

```
clean       Удалить кэшированные файлы
completion  Генерировать сценарий автозавершения для указанной оболочки
convert     Преобразовать trivy JSON отчет в другой формат
help        Справка о любой команде
server      Режим сервера
version     Показать версию
```

Сканеры

```
vuln — известные уязвимости (CVE) (по умолчанию)
config — проблемы с IAC и неправильные настройки
secret — конфиденциальная информация и секреты (по умолчанию)
license — лицензии на программное обеспечение
```

Получение подробной информации о команде

```
$ trivy команда --help
```

Примеры

Сканировать образ контейнера

```
$ trivy image inferit/msvsphere:10
```

Сканирование образа контейнера на наличие уязвимостей HIGH и CRITICAL с сохранением результата формате JSON в файл

```
$ trivy image --severity HIGH,CRITICAL -f json -o test.json inferit/msvsphere:10
```

Вывести проблемы с лицензиями

```
$ trivy image --scanners license inferit/msvsphere:10
```

Проверка конфигурации образа контейнера

```
$ trivy image --image-config-scanners misconfig inferit/msvsphere:10
```

Проверка секретов образа контейнера

```
$ trivy image --image-config-scanners secret inferit/msvsphere:10
```

Сканировать образ контейнера из архива tar

```
$ trivy image --input msvsphere.tar
```

Сканировать локальную файловую систему

```
$ trivy fs .
```

Запуск в режиме сервера

```
$ trivy server
```

Слушать на 0.0.0.0:10000

```
$ trivy server --listen 0.0.0.0:10000
```

Слушать на 0.0.0.0:10000, использовать локальную базу из /opt/trivy-local-db/

```
trivy --cache-dir /opt/trivy-local-db/ --listen 0.0.0.0:10000 server --skip-db-update
```

Сканировать образ контейнера с удалённой машины

```
trivy image --server http://192.68.10.10:10000 inferit/msvsphere:10
```

где `192.168.10.10` — IP-адрес машины, на которой запущен сервер trivy.

Сканировать файловую систему с использованием локальной базы данных (с машины, на которой запущен сервер trivy)

```
$ trivy fs --server http://localhost:4954 ./
```

Сканирование локального образа контейнера в Podman

```
$ podman images
REPOSITORY                  TAG         IMAGE ID      CREATED        SIZE
docker.io/inferit/msvsphere  latest      3a82917c452a  18 months ago  157 MB
```

```
$ trivy image 3a82917c452a
```

Для возможности сканирования локальных образов должен быть запущен `podman.socket`:

```
$ systemctl --user start podman.socket
```

Репозиторий Git

Сканирование репозитория Git на уязвимости и конфиденциальную информацию

```
$ trivy repo https://github.com/msvsphere/rpmqc
```

Сканирование ветки в репозитории на проблемы с лицензиями

```
$ trivy repo --scanners license --branch master https://github.com/msvsphere/rpmqc
```

Также можно использовать `--commit` и `--tag`.

Kubernetes

Просканировать кластер и создать простой сводный отчёт

```
$ trivy k8s --report=summary cluster
```

Просканировать кластер и вывести всю информацию о критических уязвимостях

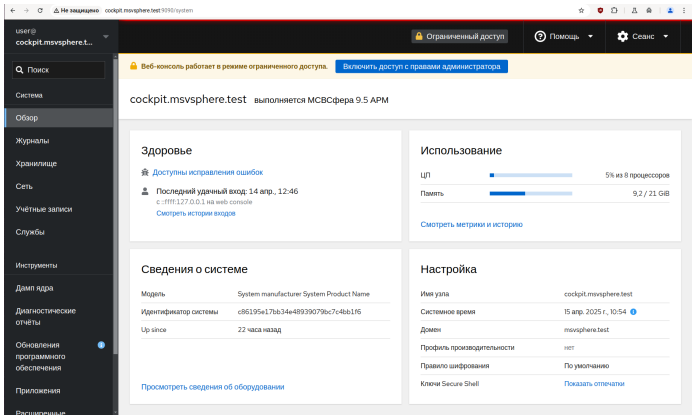
```
$ trivy k8s --report=all  --severity=CRITICAL  cluster
```

Панель управления Cockpit

Описание панели управления Cockpit

Введение

Cockpit — это панель управления для администрирования серверов и рабочих станций под управлением ОС МСВСфера. Благодаря использованию веб-интерфейса Cockpit позволяет администрировать как локальную, так и удалённые системы — для работы с панелью требуется только веб-браузер.



Основные функции

Ниже перечислены основные возможности панели управления Cockpit.

- Мониторинг и диагностика:
  - просмотр сведений о системе и конфигурации оборудования;
  - мониторинг использования ресурсов: центрального процессора, оперативной памяти, дискового пространства и сетевых интерфейсов;
  - просмотр и поиск по системным журналам;
  - диагностика ошибок подсистемы SELinux.
- Администрирование:
  - настройка системного времени;
  - ввод компьютера в домен Active Directory или FreeIPA;
  - управление учётными записями пользователей и группами, а также политиками паролей;
  - управление системными службами: (пере)запуск и остановка, настройка автоматического запуска, просмотр журналов;
  - управление обновлениями программного обеспечения;
  - управление сетевыми интерфейсами и настройками брандмауэра;

- управление локальными (LVM, RAID) и сетевыми (NFS, iSCSI) хранилищами;
- управление виртуальными машинами (Libvirt) и контейнерами (Podman);
- настройка аварийного дампа ядра (kdump).

Также следует отметить, что Cockpit является модульной системой, что делает возможной разработку собственных расширений — этот процесс подробно описан в официальной документации проекта.

Установка и настройка Cockpit

Установка

Для установки панели управления Cockpit выполните следующую команду:

```
$ sudo dnf install cockpit
```

Затем, запустите службу `cockpit.socket`:

```
$ sudo systemctl enable --now cockpit.socket
```

После запуска интерфейс системы управления будет доступен по локальному адресу https://localhost:9090/. В случае обращения к удалённому компьютеру укажите вместо `localhost` его IP-адрес или доменное имя.

В зависимости от настроек вашего брандмауэра, также может потребоваться открыть доступ к порту, на котором запущена панель управления Cockpit:

```
$ sudo firewall-cmd --add-service=cockpit --permanent
success

$ sudo firewall-cmd --reload
success
```

Создание диагностических отчётов

Введение

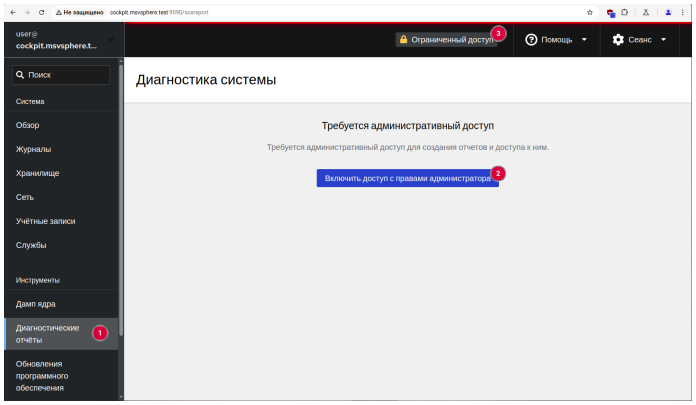
В состав панели управления Cockpit входит модуль для формирования диагностических отчётов, которые затем могут быть переданы для исследования в службу технической поддержки.

Для активации этого модуля предварительно необходимо установить пакет `sos`:

```
$ sudo dnf install sos
```

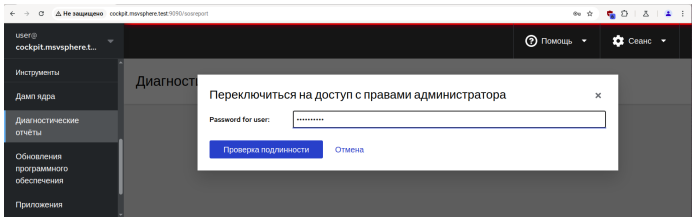
Создание отчёта

Для создания диагностического отчёта войдите в систему управления Cockpit и в левом навигационном меню выберите пункт «**Диагностические отчёты**» (отмечен цифрой 1 на снимке экрана).

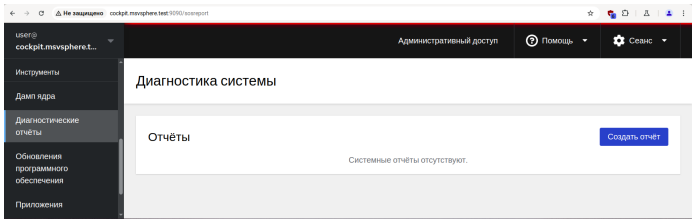


Если вы вошли в панель управления, используя учётную запись непривилегированного пользователя, то получите привилегии администратора либо нажав кнопку «**Включить доступ с правами администратора**» (отмечена цифрой 2 на снимке экрана выше), либо кнопку «**Ограниченный доступ**» (отмечена цифрой 3 на снимке экрана выше).

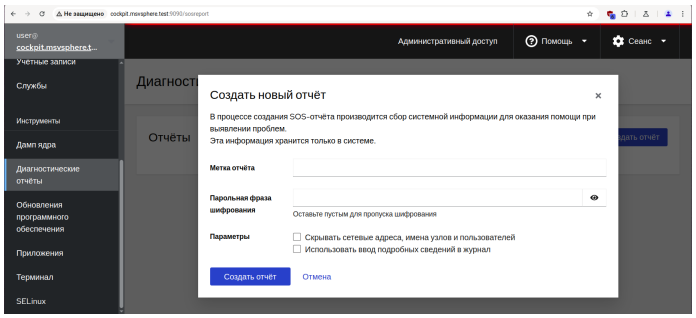
В открывшейся форме необходимо ввести свой пароль и нажать кнопку «**Проверка подлинности**».



После этого на экране появится список ранее созданных отчётов (в данном примере он пустой) и кнопка «**Создать отчёт**».



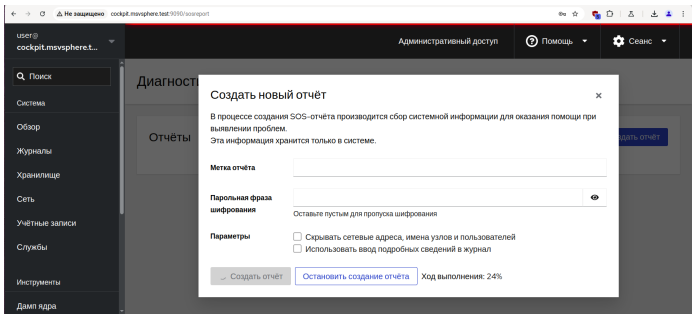
После нажатия на кнопку появится соответствующая форма, в которой вы можете настроить параметры создания отчёта.



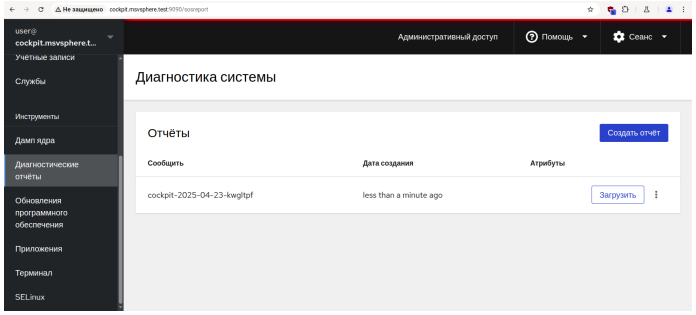
На данный момент для конфигурации доступны следующие опции:

- В поле «**Метка отчёта**» вы можете ввести идентификатор запроса в службе технической поддержки. Если вы не знаете его, оставьте поле пустым.
- В поле «**Парольная фраза шифрования**» вы можете указать пароль, который затем потребуется ввести для открытия файла отчёта. Защита файла отчёта паролем не является обязательным требованием, однако, рекомендуется в случае передачи файла по незащищённым каналам.
- Флажок «**Скрывать сетевые адреса, имена узлов и пользователей**» включает режим обезличивания этих данных в отчёте. Следует отметить, что пароли, ключи и другие приватные данные в любом случае не будут включены в отчёт.
- Флажок «**Использовать ввод подробных сведений в журнал**» включает дополнительную отладочную информацию в отчёт. Используйте его по запросу инженера технической поддержки.

После нажатия на кнопку «**Создать отчёт**» запустится процедура генерации, которая может занять несколько минут в зависимости от размера файлов журналов.



После завершения процедуры форма создания отчёта будет автоматически закрыта, а в списке отчётов появится новая запись, в данном примере — `cockpit-2025-04-23-kwgltpf`.



Для генерации идентификатора отчёта используется шаблон `HOSTNAME-YYYY-MM-DD-ID`, где:

- `HOSTNAME` — имя компьютера;
- `YYYY` — текущий год;
- `MM` — текущий месяц;
- `DD` — текущий день;
- `ID` — уникальный идентификатор.

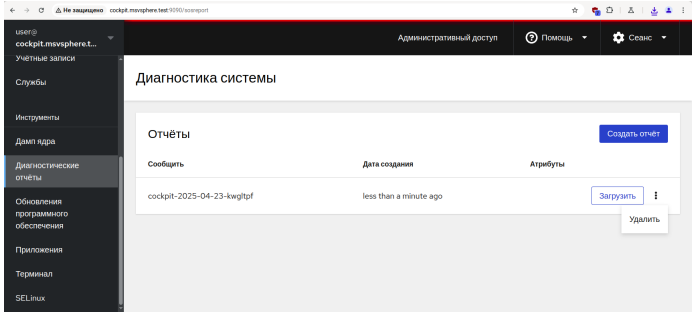
Таким образом, отчёт с именем `cockpit-2025-04-23-kwgltpf` был создан на компьютере *cockpit* 23 апреля 2025 года. Если в форме настройки параметров отчёта была задана «**Метка отчёта**», то эта информация также попадёт в идентификатор созданного отчёта.

Созданный отчёт можно скачать нажав на кнопку «**Загрузить**», после этого его необходимо прикрепить к заявке, созданной на портале технической поддержки, либо передать в службу поддержки иным согласованным способом.

Скачанный файл будет иметь префикс `sosreport-`, если файл не защищён паролем и префикс `secured-sosreport-`, если пароль был установлен. Соответственно, в нашем примере для отчёта `cockpit-2025-04-23-kwgltpf` будет скачан файл `sosreport-cockpit-2025-04-23-kwgltpf.tar.xz`.

Удаление отчёта

Удалить файл отчёта из системы можно нажав на кнопку меню «⊠» справа от кнопки «**Загрузить**» и выбрав там пункт меню «**Удалить**».



Просмотр данных отчёта

Файл отчёта представляет собой tar-архив, сжатый архиватором `xz`, который содержит следующую информацию:

- настройки загрузчика и конфигурационный файл ядра системы;
- конфигурационные файлы различных компонентов из каталога `/etc`;
- список загруженных модулей ядра, запущенных процессов, открытых файловых дескрипторов и т.д.;
- конфигурацию сетевых адаптеров и информацию об оборудовании;
- системные журналы различных компонентов системы.

Для распаковки файлов отчёта в текущий каталог выполните следующую команду (замените имя файла на реальное):

```
$ tar -xJvf sosreport-cockpit-2025-04-23-kwgltpf.tar.xz
```

Если при создании отчёта был установлен пароль, то перед распаковкой необходимо расшифровать файл используя следующую команду (замените имена файлов на реальные):

```
$ gpg -d -o secured-sosreport-cockpit-2025-04-24-azytuxv.tar.xz \
    secured-sosreport-cockpit-2025-04-24-azytuxv.tar.xz.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
```

После ввода пароля в текущем каталоге будет создан архив `.tar.xz` с указанным именем, который затем можно будет распаковать приведённой выше командой.

Настройка мультитерминального режима

Введение

Мультитерминальный режим — это особая конфигурация компьютера и операционной системы, которая позволяет обеспечить одновременную работу нескольких пользователей за одним компьютером.

В состав операционной системы МСВСфера входит расширение для панели управления Cockpit, которое предоставляет графический интерфейс для настройки работы в мультитерминальном режиме.

Аппаратные требования

Для работы в мультитерминальном режиме необходима следующая конфигурация компьютера:

- отдельная видеокарта для каждого рабочего места. Оптимальный вариант — установить дополнительную видеокарту в компьютер со встроенной видеокартой;
- отдельная клавиатура для каждого рабочего места. Рекомендуется использовать разные модели клавиатур чтобы их было проще отличать друг от друга в интерфейсе управления;
- отдельная мышка (трекбол, тачпад и т.п.) для каждого рабочего места. Как и в случае с клавиатурами, не рекомендуется использование устройств одинаковой модели;
- если требуется обеспечить вывод звука для каждого рабочего места, то вам также потребуется отдельная звуковая карта. В качестве альтернативы можно выводить звук на монитор через HDMI.

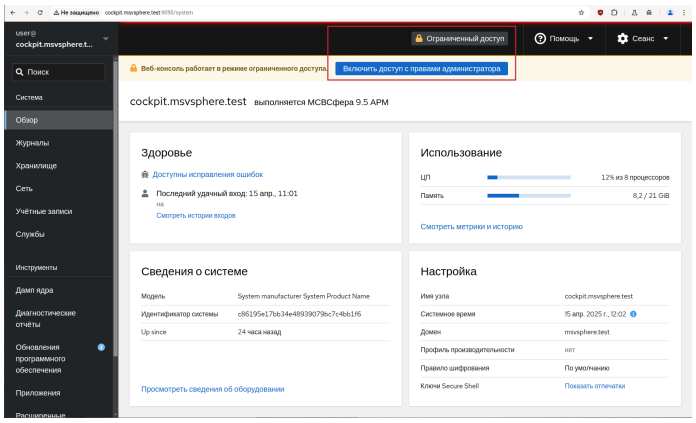
Установка расширения для Cockpit

Для настройки мультитерминальной системы установите соответствующее расширение для панели управления Cockpit:

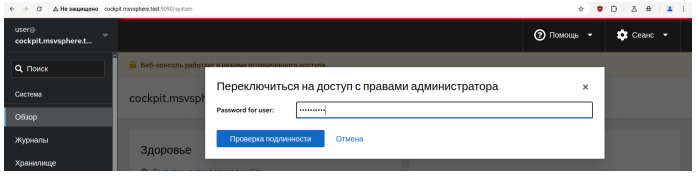
```
$ sudo dnf install cockpit-msvsphere-multi-seat
```

Настройка дополнительного рабочего места

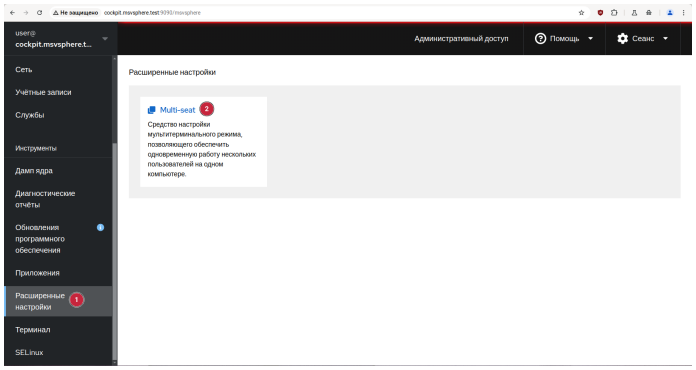
Для настройки мультитерминального режима требуются привилегии администратора: войдите в панель управления Cockpit, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником.



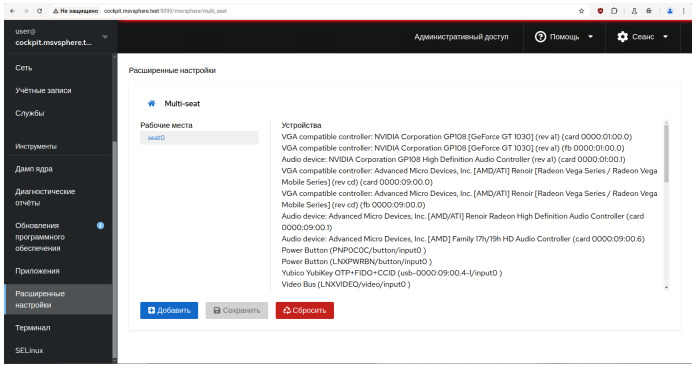
В открывшейся форме необходимо ввести свой пароль и нажать кнопку «Проверка подлинности».



После этого в левой панели откройте страницу «Расширенные настройки» (обозначена цифрой 1 на снимке экрана), там перейдите по ссылке «Multi-seat» (обозначена цифрой 2 на снимке экрана).

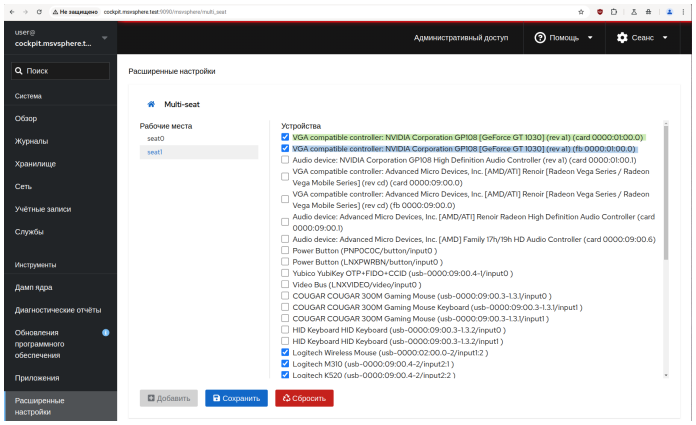


По умолчанию в системе настроено только одно рабочее место (`seat0`), за которым закреплены все доступные устройства. Для добавления ещё одного рабочего места нажмите кнопку «Добавить».

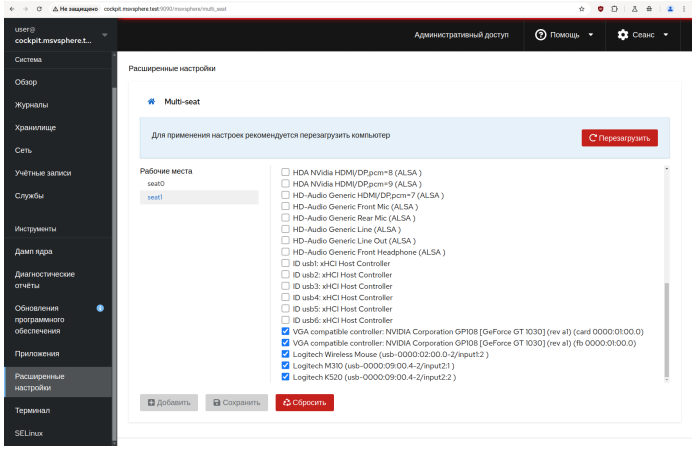


В результате будет создано дополнительное рабочее место (`seat1`) и вам потребуется назначить ему соответствующие устройства ввода-вывода, используя флажки в списке устройств. Минимально работоспособная конфигурация должна включать в себя видеокарту, клавиатуру и мышку (трекбол, тачпад и т.п.).

Также обратите внимание, что для видеокарты в списке отображаются два устройства: тип `card` (выделено зелёным маркером на снимке экрана) и тип `fb` (выделено синим маркером на снимке экрана) — вам необходимо закрепить оба устройства одной видеокарты за рабочим местом.



После завершения конфигурации устройств нажмите кнопку «Сохранить» — система выдаст всплывающее сообщение об успешном добавлении рабочего места и предложит перезагрузить компьютер для активации изменений.

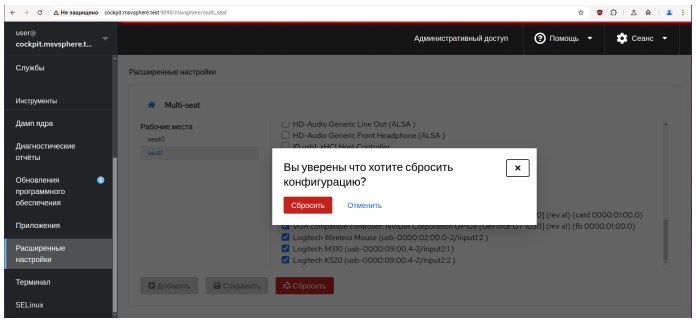


После перезагрузки на каждом из рабочих мест будет запущена собственная графическая сессия и пользователи смогут войти в систему используя свои учётные записи.

Возврат к настройкам по умолчанию

Для возврата системы к базовой конфигурации с одним рабочим местом войдите в панель управления Cockpit, получите привилегии администратора, перейдите на страницу «Расширенные настройки» и выберите там расширение «Multi-seat» как было описано в предыдущем разделе.

На странице расширения нажмите на кнопку «Сбросить», появится соответствующее окно для подтверждения операции.



После подтверждения система выдаст всплывающее сообщение об успешном сбросе конфигурации и предложит перезагрузить компьютер. После перезагрузки система вернётся в исходное состояние с одним рабочим местом.

Расширение USBGuard для Cockpit

Введение

Расширение USBGuard для Cockpit работает на основе программной платформы USBGuard и помогает защитить компьютер от мошеннических USB-устройств, внедряя базовые возможности белого и чёрного списков на основе атрибутов USB-устройств.

Установка расширения для Cockpit

Установите расширение для панели управления `Cockpit` с помощью следующей команды:

```
$ sudo dnf install cockpit-msvsphere-usbguard
```

Запустите сервис `usbguard` с помощью следующей команды:

```
$ sudo systemctl enable --now usbguard.service
```

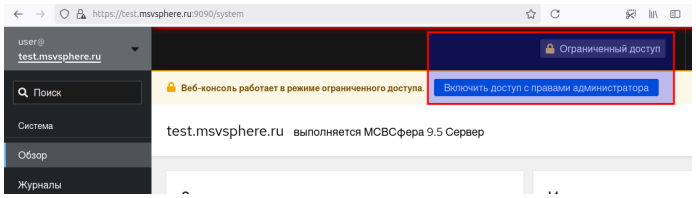
По умолчанию USBGuard блокирует USB-устройства, не соответствующие правилам. После запуска сервиса `usbguard` в «белый» список добавятся все текущие подключённые устройства. Все новые подключённые USB-устройства будут блокироваться.

Настройка

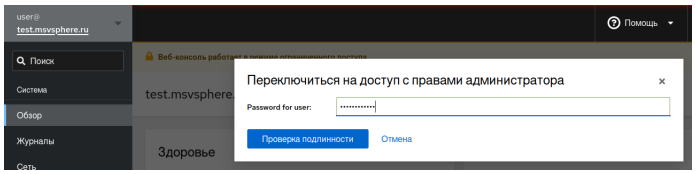
Настройка пользователей

По умолчанию для работы с расширением USBGuard требуются права пользователя `root` или же пользователь должен состоять в группе `wheel`. Изменить настройки и добавить нового пользователя/группу может пользователь с правами администратора. Для этого выполните следующие действия.

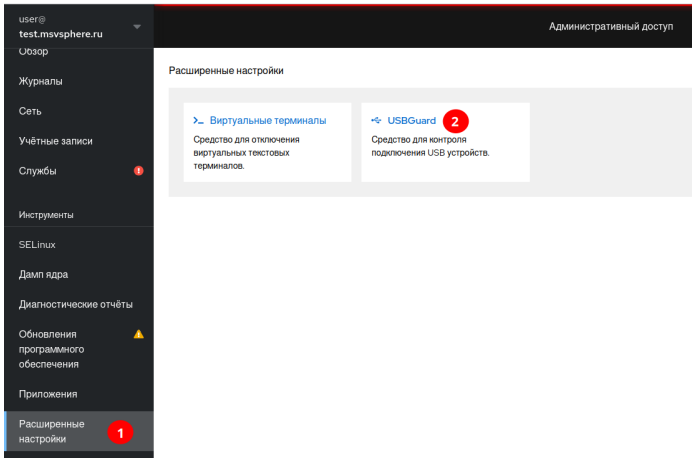
- Войдите в панель управления `Cockpit`. На странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки выделены красным.



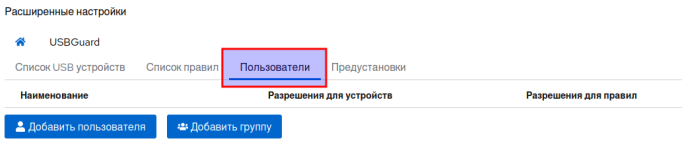
- В открывшейся форме укажите свой пароль и нажмите кнопку «Проверка подлинности».



- После этого в левой панели откройте страницу «Расширенные настройки» (обозначена цифрой 1 на снимке экрана), там перейдите по ссылке «USBGuard» (обозначена цифрой 2 на снимке экрана).

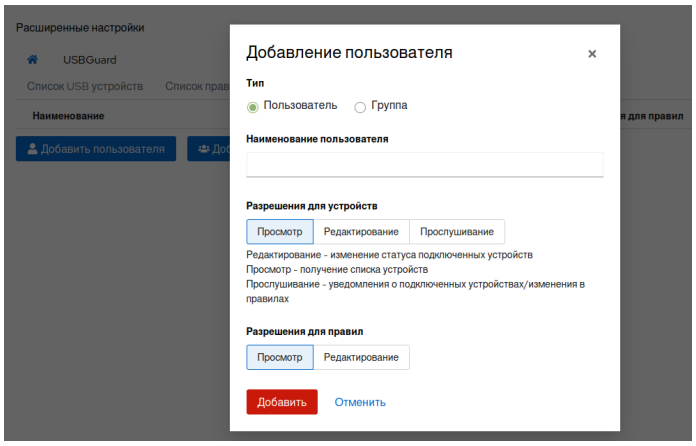


- Перейдите на вкладку «Пользователи».



Добавление пользователя/группы

Для добавления пользователя или группы нажмите кнопку «Добавить пользователя» или «Добавить группу». В открывшемся окне вы можете указать создаваемый тип: «Пользователь» или «Группа», наименование и необходимые разрешения для устройств и правил.



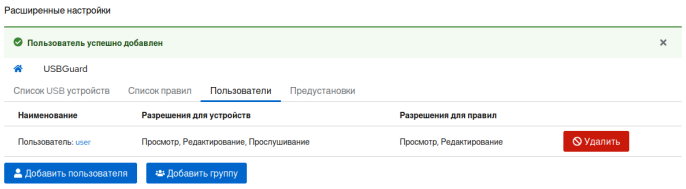
Разрешения для устройств включают следующие значения.

- Просмотр (list) — получать список подключённых USB-устройств и их атрибуты.
- Редактирование (modify) — изменять состояние авторизации USB-устройств (блокировка/разблокировка), включая постоянные изменения (т.е. изменение конкретных правил или устройств в политике).
- Прослушивание (listen) — получать уведомления о подключении устройств и изменении политики устройств.

Разрешения для правил включают следующие значения.

- Просмотр (list) — получать список правил.
- Редактирование (modify) — изменять и удалять правила.

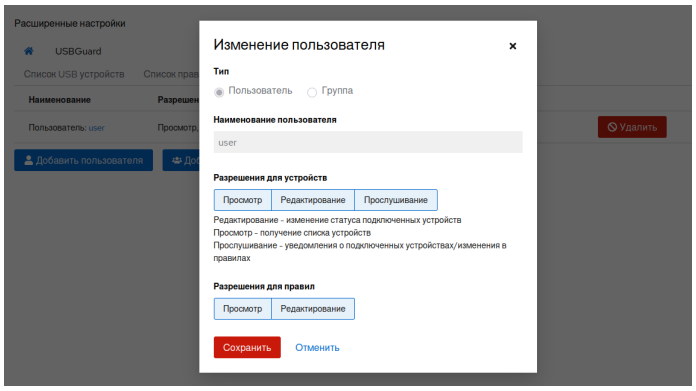
После добавления пользователя сервис `usbguard` будет перезапущен.



В дальнейшем, в зависимости от разрешений, пользователь может получить доступ к интерфейсу USBGuard без повышения привилегий.

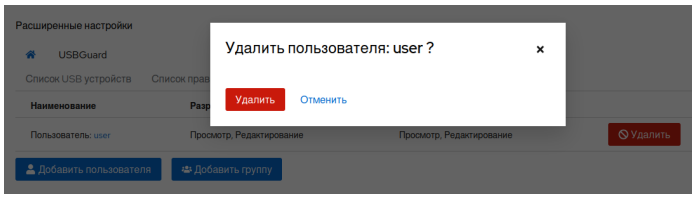
Изменение пользователя/группы

Для изменения пользователя/группы нажмите на него в таблице. В открывшемся окне вы можете изменить требуемые разрешения. После сохранения изменений сервис `usbguard` будет перезапущен.



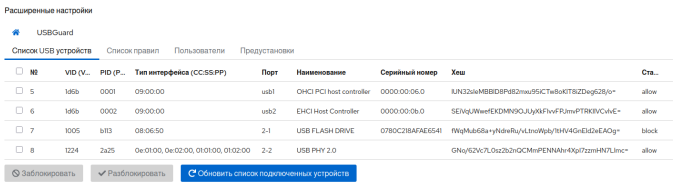
Удаление пользователя/группы

Для удаления пользователя/группы нажмите на кнопку «Удалить» в таблице пользователей и подтвердите действие. После удаления сервис `usbguard` будет перезапущен.



Список USB-устройств

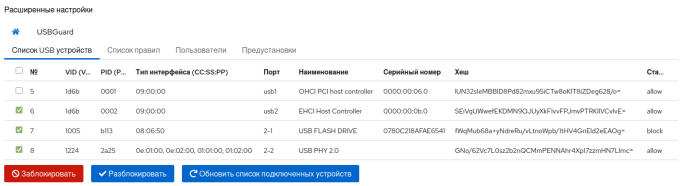
На вкладке отображается список текущих подключённых USB устройств.



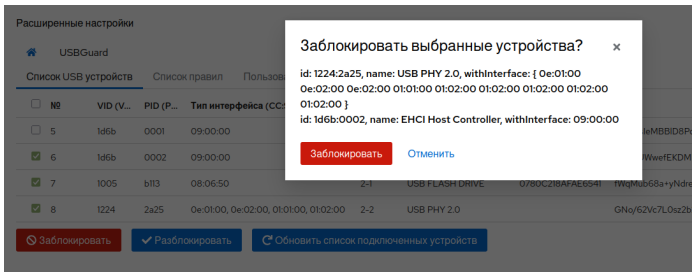
В столбце «Статус» отображается текущее состояние USB-устройства:

- allow — разрешено;
- block — заблокировано.

Для редактирования состояния USB-устройства выделите строку с устройством. В зависимости от текущего статуса устройства станут доступны кнопки «Разблокировать/»Заблокировать».



Нажмите на кнопку выбранного действия и подтвердите его в открывшемся окне.

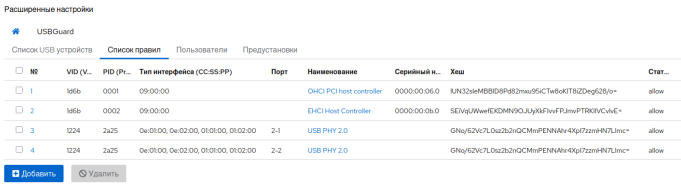


После подтверждения будет добавлено соответствующее правило для выбранных устройств.

Также в данном разделе доступна кнопка «Обновить список подключённых устройств» для обновления списка после подключения или удаления USB-устройств.

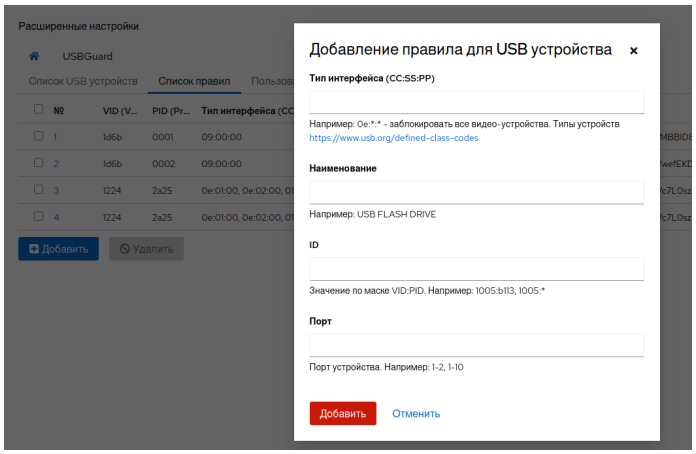
Список правил

На вкладке «Список правил» отображается список текущих правил для USB-устройств, основанный на их атрибутах.



Добавление правила

Для добавления правила нажмите на кнопку «Добавить». Откроется окно с формой добавления.



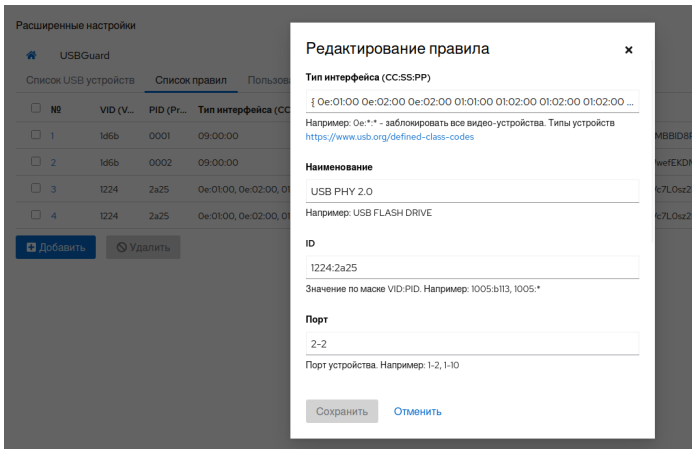
Доступны следующие атрибуты.

- Тип интерфейса (CC:SS:PP) — тип интерфейса указывается как три 8-битных числа в шестнадцатеричном формате, разделённых двоеточием (`CC:SS:PP`). Числа обозначают класс интерфейса (CC), подкласс (SS) и протокол (PP). Пример списка классов. Вместо номера подкласса и протокола можно использовать символ *, чтобы выбрать все подклассы или протоколы. Также есть возможность указать несколько типов, при этом значения должны быть разделены пробелом и обрамлены фигурными скобками. Например: `{ 03:01:01 03:00:00 }`.
- Наименование — название устройства.
- ID — значение по маске `VID:PID` (`<ID вендора>:<ID продукта>`). Формат — 16-битные числа в шестнадцатеричном формате. В значении может быть указан символ *, чтобы выбрать все значения.
- Порт — USB-порт устройства.
- Серийный номер устройства.
- Хеш устройства.
- Статус — может принимать одно из значений: `block` — заблокировать устройство, `allow` — разрешить, `reject` — удалить устройство из системы.

Правило вступает в силу после подключения USB-устройства. Таким образом, если устройство было подключено на момент создания правила, то правило начнёт действовать после его переподлючения.

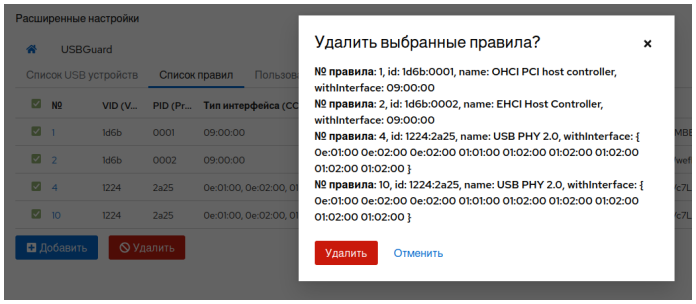
Изменение правила

Для изменения правила нажмите на его номер в столбце «№» или на его имя в столбце «Наименование». Откроется окно редактирования правила.



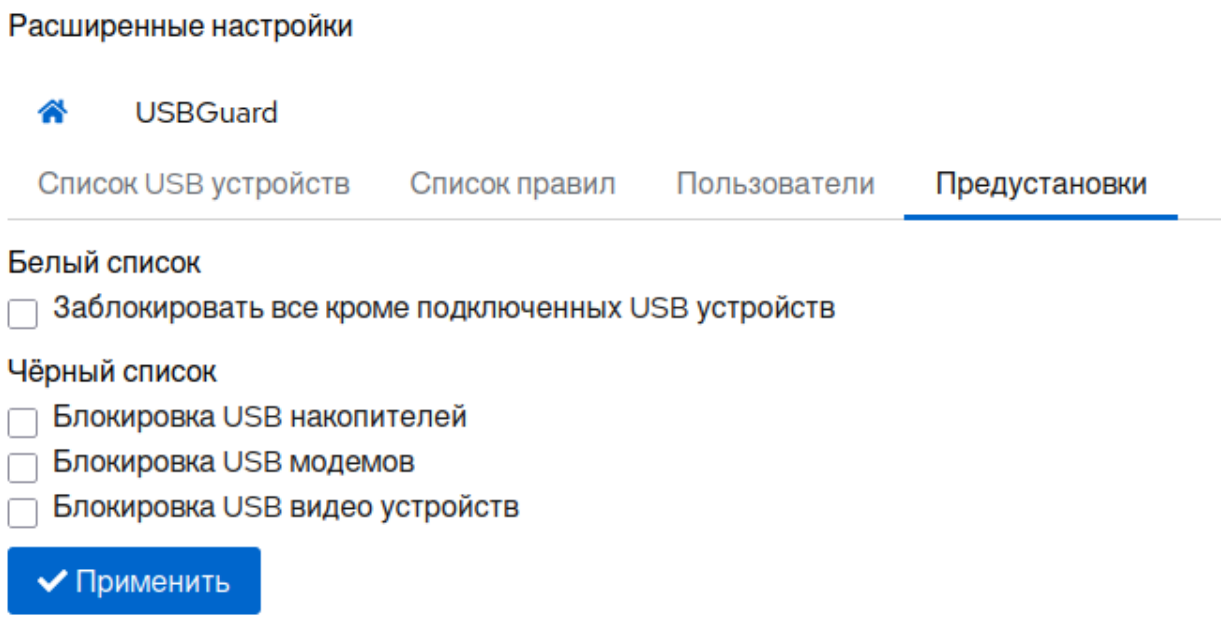
Удаление правил

Для удаления правила выберите его в списке и нажмите на кнопку «Удалить». В открывшемся окне подтвердите свои действия.



Предустановки

В разделе «Предустановки» вы можете отключить правила, а также добавить общие правила для некоторых классов USB-устройств.



Белый список

- `Заблокировать все кроме подключённых USB-устройств` — все текущие правиле будут удалены, затем на основе списка подключённых устройств будут сгенерированы и добавлены новые правила. Также в конфигурационном файле USBGuard `%sysconfdir%/usbguard/rules.conf` значение `ImplicitPolicyTarget` (статус для новых устройств, не подходящих под текущие правила) будет изменено на `block` (по умолчанию).

Чёрный список

- `Блокировка USB накопителей` — добавление правила с типом интерфейса `08:*:*`.
- `Блокировка USB модемов` — добавление правила с типом интерфейса `02:*:*`
- `Блокировка USB видео устройств` — добавление правила с типом интерфейса `0e:*:*`

Расширение Bootloader для Cockpit

Введение

Расширение Bootloader для Cockpit предоставляет интерфейс настройки параметров загрузчика операционной системы.

Установка расширения для Cockpit

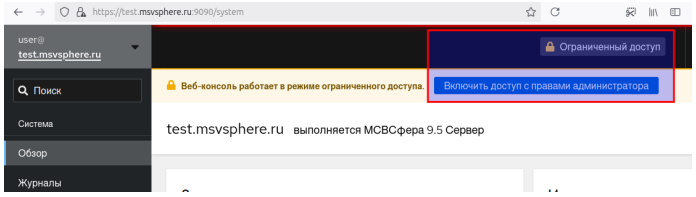
Установите соответствующее расширение для панели управления `Cockpit`:

```
$ sudo dnf install cockpit-msvsphere-bootloader
```

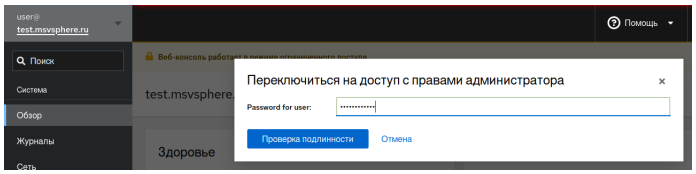
Настройка

Для изменения параметров загрузчика требуются привилегии администратора:

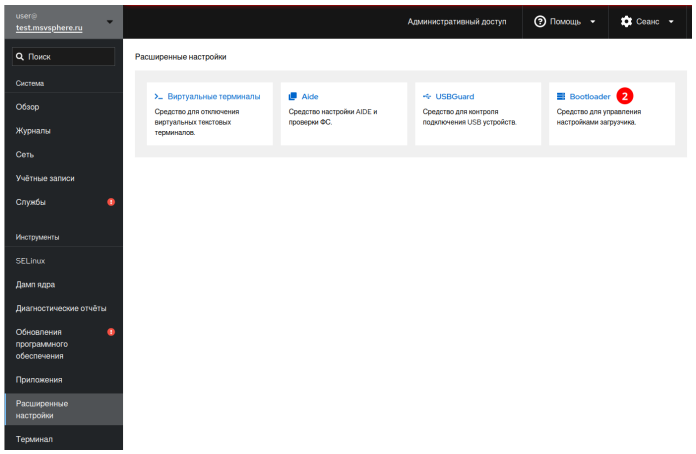
- войдите в панель управления `Cockpit`, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником:



- В открывшейся форме необходимо ввести свой пароль и нажать кнопку «Проверка подлинности»:



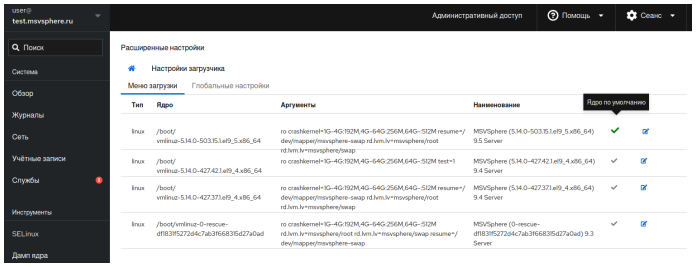
- После этого в левой панели откройте страницу «Расширенные настройки» (обозначена цифрой *1* на снимке экрана), там перейдите по ссылке «Bootloader» (обозначена цифрой *2* на снимке экрана):



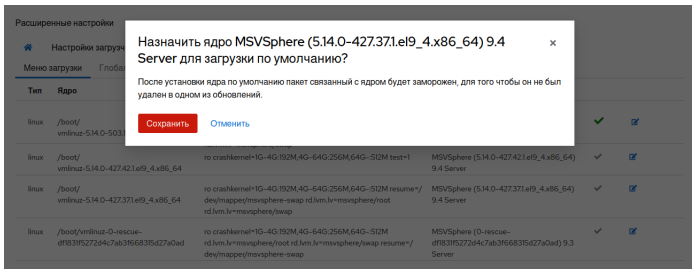
Вкладка «Меню загрузки»

На данной вкладке будут отображены записи из меню загрузки.

Ядро, загруженное по умолчанию, будет отмечено иконкой зелёного цвета.

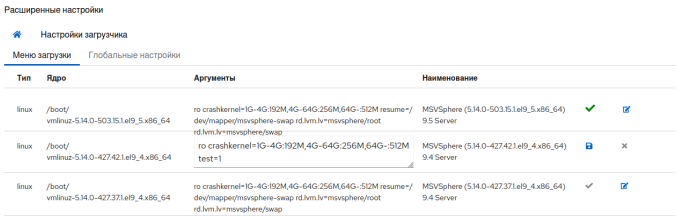


Для того чтобы изменить загружаемое ядро по умолчанию, нажмите иконку на соответствующей записи. После нажатия во всплывающем окне подтвердите свой выбор.



После установки нового ядра для загрузки по умолчанию пакет, связанный с ядром, будет «заморожен» (с помощью `dnf versionlock`), чтобы он не был удалён из системы при последующих обновлениях ядра.

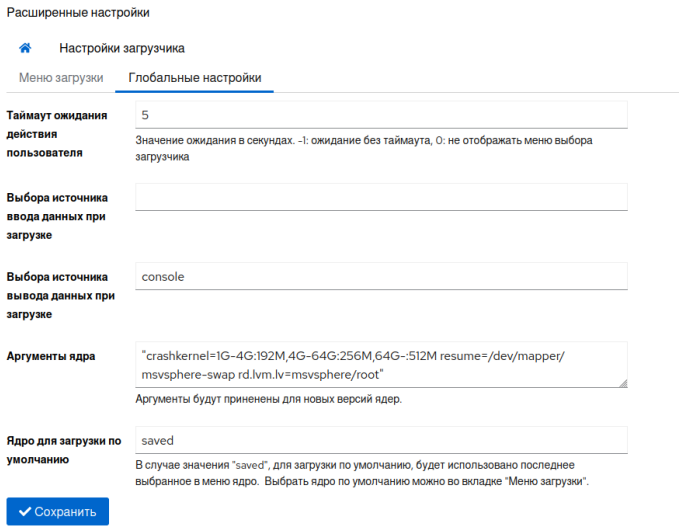
Также в этом разделе есть возможность изменения аргументов ядра.



После изменения значений в текстовом поле нажмите иконку «сохранить».

Вкладка «Глобальные настройки»

В данном разделе представлена форма, позволяющая изменять некоторые глобальные настройки загрузчика путём модификации файла `/etc/default/grub`



Текущие настройки:

- **`Таймаут ожидания действия пользователя` — время ожидания действия в секундах.**

  Значение `-1` — ожидание выбора без таймаута, `0` — не отображать меню выбора загрузчика.

- **`Выбора источника ввода данных при загрузке` — терминальное устройство ввода. Здесь вы можете выбрать несколько устройств, разделенных пробелами.**

  Допустимые названия терминалов ввода зависят от платформы, но могут включать `console` (консоль собственной платформы), `serial` (последовательный терминал), `serial_<порт>` (последовательный терминал с явным выбором порта), `at_keyboard` (клавиатура PC AT) или `usb_keyboard` (USB-клавиатура). По умолчанию используется собственный терминальный ввод платформы.

- **`Выбора источника вывода данных при загрузке` — терминальное устройство вывода. Здесь вы можете выбрать несколько устройств, разделенных пробелами.**

  Допустимые названия выходных данных терминала зависят от платформы, но могут включать `console` (консоль собственной платформы), `serial` (последовательный терминал), `serial_<порт>` (последовательный терминал с явным выбором порта), `gfxterm` (вывод в графическом режиме), `vga_text` (текст VGA вывод), `mda_text` (вывод текста MDA), `morse` (кодировка Морзе с использованием системного звукового сигнала) или `spkmodem` (простой протокол передачи данных с использованием системного динамика).

- `Аргументы ядра` — будут использоваться как аргументы по умолчанию при обновлении/установке новых ядер.

- **`Ядро для загрузки по умолчанию` — в случае значения `saved` для загрузки по умолчанию будет использовано последнее выбранное в меню ядро.**
  Выбрать ядро для загрузки по умолчанию также можно во вкладке «Меню загрузки».

Подключение к домену

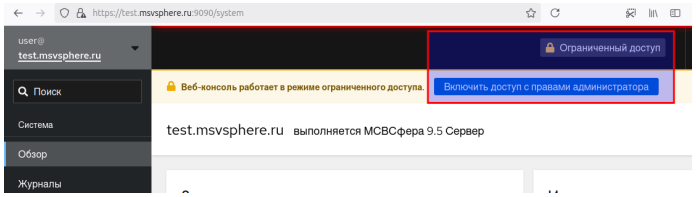
Введение

Cockpit предоставляет возможность подключения к домену.

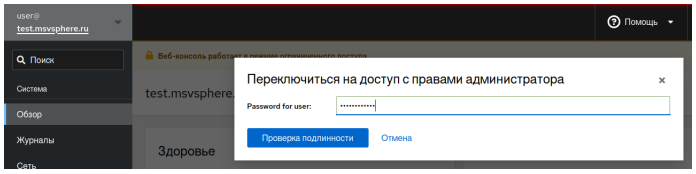
Настройка

Для подключения требуются привилегии администратора.

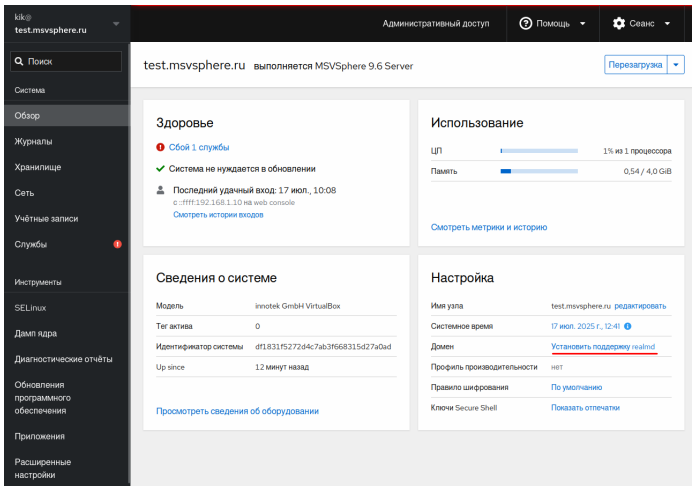
- Войдите в панель управления `Cockpit`, на странице «Обзор» нажмите на кнопку «Включить доступ с правами администратора» или на кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником:



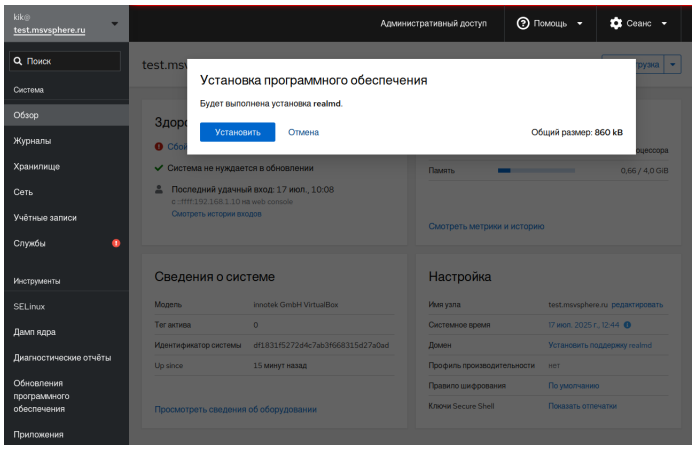
- В открывшейся форме укажите пароль и нажмите на кнопку «Проверка подлинности»:



- После этого в разделе «Настройка» (пункт «Домен») нажмите на ссылку «Установить поддержку realmd»:

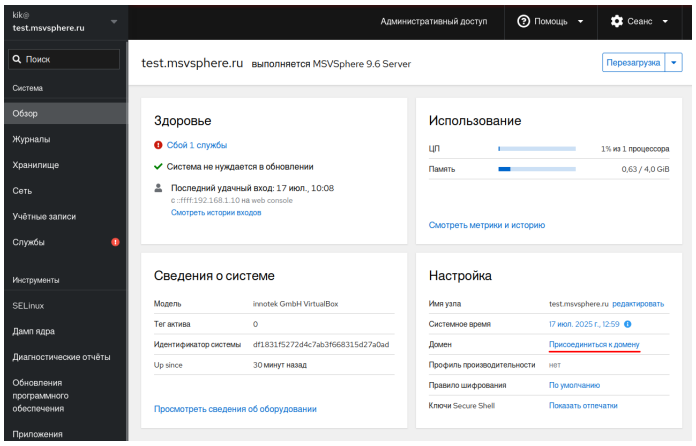


- Затем выполните установку, нажав на кнопку «Установить»:



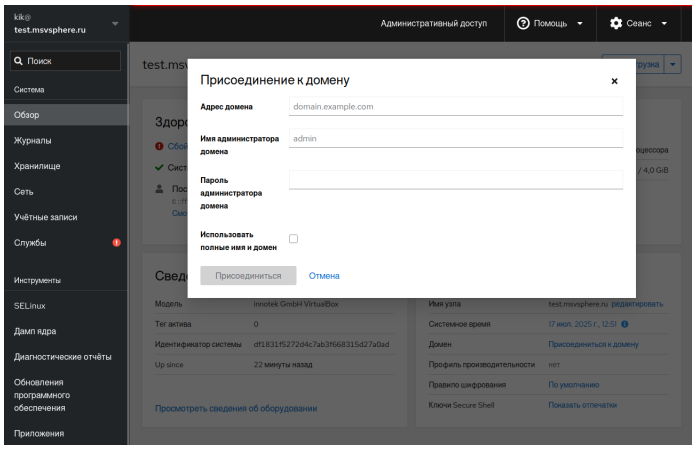
Подключение к домену

После установки пакета `realmd` вы увидите окно с формой для подключения к домену. Или же в разделе «Настройка» (пункт «Домен») нажмите на ссылку «Присоединится к домену».

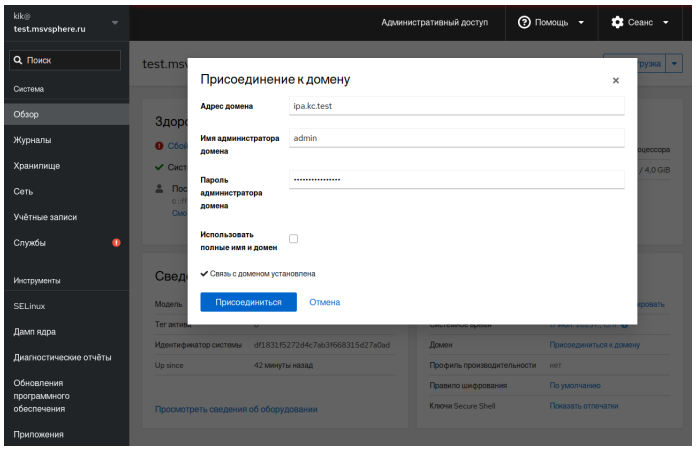


Заполните поля формы:

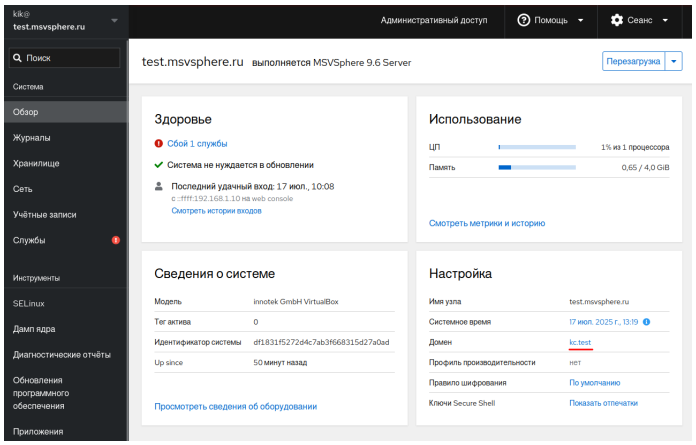
- `Адрес домена` — адрес сервера, управляющего доменом.
- `Имя администратора домена` — по умолчанию `admin`.
- `Пароль администратора домена`.
- `Использовать полные имя и домен` — настройка позволяет разрешить аутентификацию по коротким именам (без указания доменного суффикса в формате `user@domain.ru`). Например, `user`. По умолчанию используются короткие имена.



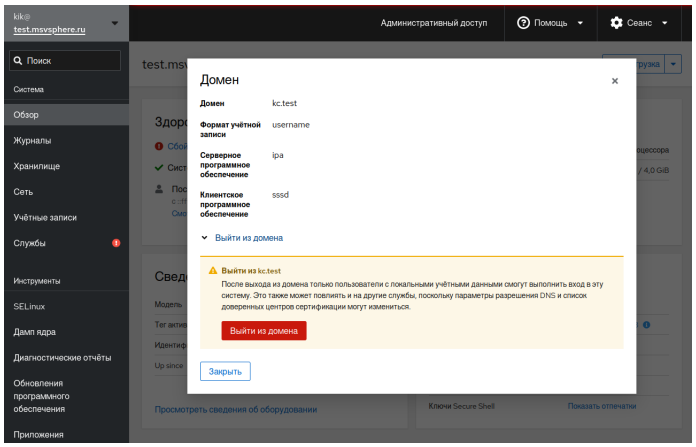
После заполнения поля `Адрес домена` выполняется проверка доступности сервера, в случае ошибки будет выведено сообщение. Если подключение прошло успешно, кнопка «Присоединиться» станет активной.



После нажатия на кнопку «Присоединиться» выполняется попытка подключения к серверу домена. В случае успешного подключения к домену в разделе «Настройка» (пункт «Домен») будет указан подключённый домен.



При нажатии на адрес домена откроется окно, в котором указаны детали подключения и предоставлена возможность отключиться от домена.



Расширение Aide для Cockpit

Введение

AIDE (Advanced Intrusion Detection Environment) — это инструмент для обнаружения вторжений на основе анализа изменений в файловой системе. Он работает по принципу системы обнаружения вторжений на основе хоста (HIDS) и помогает администраторам выявлять несанкционированные изменения в файлах и каталогах.

Установка расширения для Cockpit

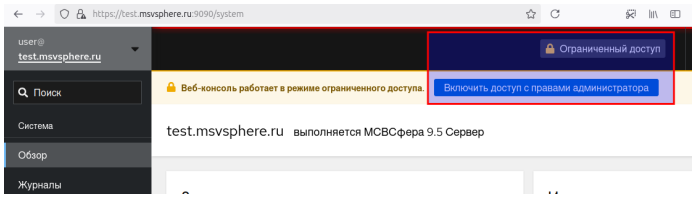
Установите соответствующее расширение для панели управления `Cockpit` с помощью следующей команды:

```
$ sudo dnf install cockpit-msvsphere-security-audit
```

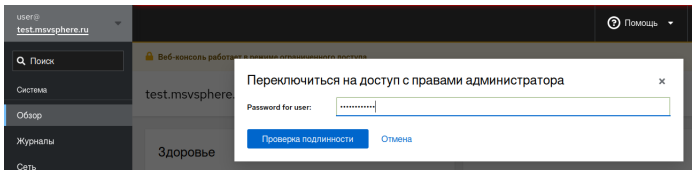
Настройка

Для использования расширения требуются привилегии администратора:

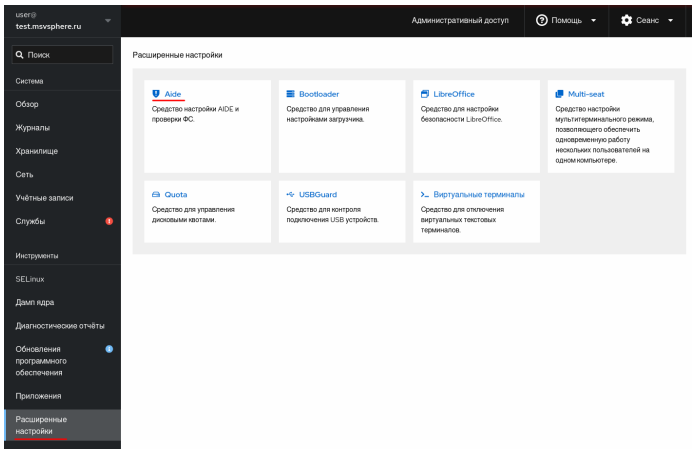
- войдите в панель управления `Cockpit`, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником.



- В открывшейся форме укажите свой пароль и нажмите на кнопку «Проверка подлинности».

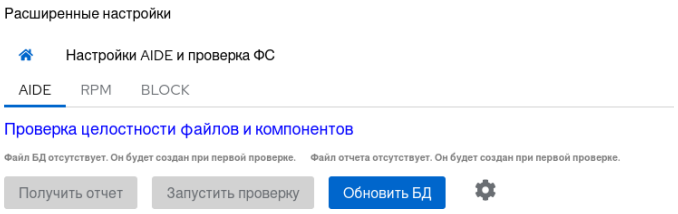


- После этого в левой панели откройте вкладку «Расширенные настройки», там перейдите по ссылке «Виртуальные терминалы».

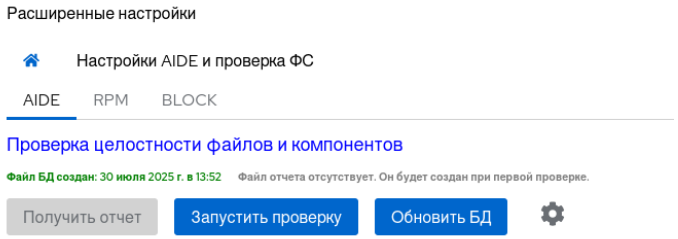


Вкладка «Aide»

После первоначальной установки необходимо инициализировать базу данных Aide. Инициализация заключается в создании базы данных (снимка) всех файлов и каталогов сервера. Для этого нажмите на кнопку «Обновить БД».



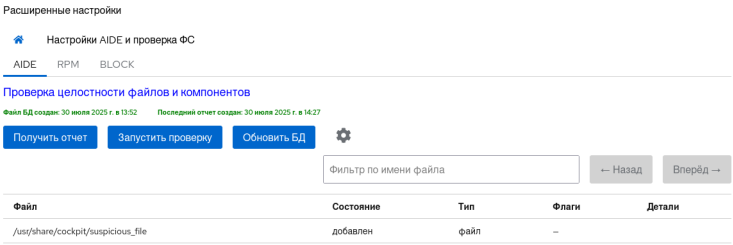
После инициализации базы данных появится возможность запустить проверку, для этого нажмите на соответсвующую кнопку.



В случае, если в системе произошли изменения по сравнению со «снимком» системы, будет выведен соответствующий результат в виде таблицы. В таблице будут указаны следующие данные:

- Файл — путь к файлу/каталогу.
- Состояние — добавлен/изменен/удалён.

- Тип — файл или каталог.
- Флаги — при наведении курсора на столбец будет показана детальная информация.
- Детали — дополнительная информация.



Список флагов:

- p — права доступа;
- i — номер inode;
- n — имя ссылки;
- u — владелец файла;
- g — группа файла;
- s — размер файла;
- b — количество блоков;
- m — время изменения ($mtime$);
- a — время доступа ($atime$);
- c — изменение inode ($ctime$).

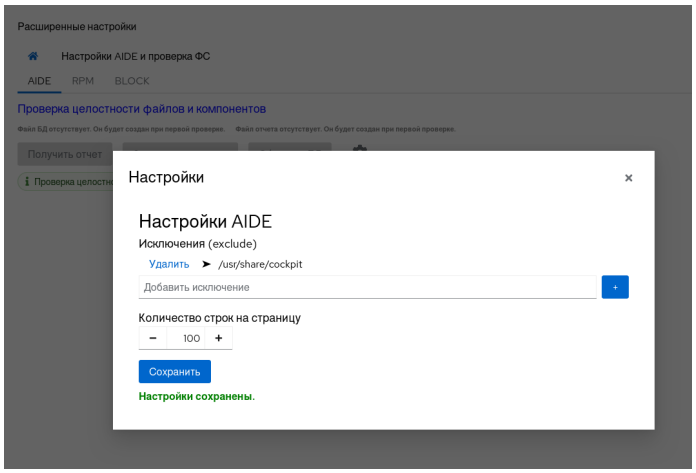
Настройки Aide

Вы можете настроить вывод требуемым образом. Для этого перейдите в «Настройки Aide», нажав на иконку «Шестеренка».

Здесь вы можете настроить следующие параметры вывода:

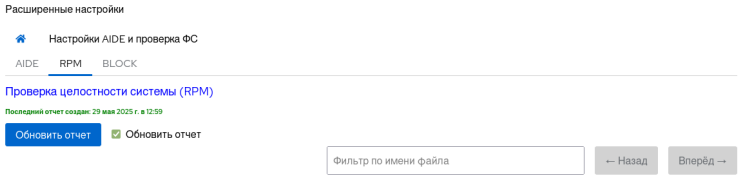
- Исключения — можно указать путь или наименование файла, который будет отфильтровываться в итоговой таблице.
- Количество строк, выводимых на странице.

Настройки применяются после сохранения.



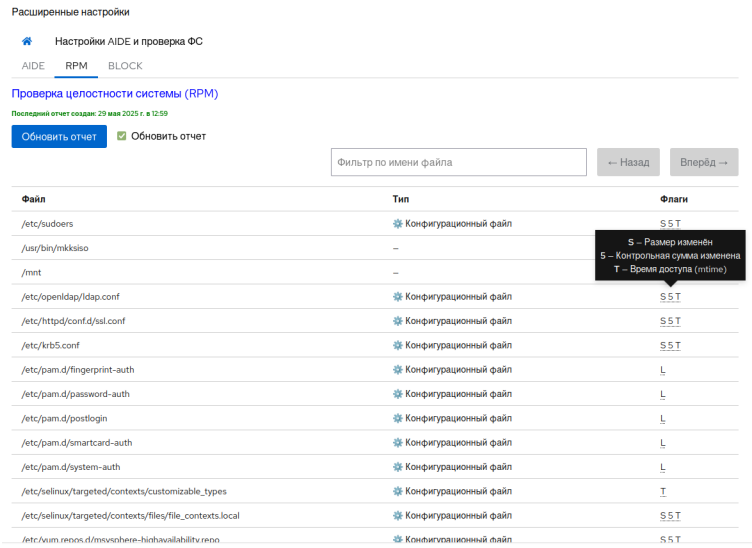
Вкладка «Rpm»

На данной вкладке вы можете запустить проверку целостности системы или просмотреть результат последней проверки. При нажатии на кнопку «Обновить отчёт» будет выполнена команда `rpm --all --verify`.



После окончания выполнения команды результат будет выведен в таблице. Будут указаны следующие данные:

- Файл — полный путь к файлу.
- Тип — тип файла (если известен).
- Флаги — при наведении на столбец будет показана детальная информация.



Список флагов:

- *S* — размер изменён;
- *M* — время модификации изменено;
- *5* — контрольная сумма изменена;
- *D* — номера major/minor изменены;
- *L* — символическая ссылка изменилась;
- *U* — пользователь изменён;
- *G* — группа изменена;
- *T* — время доступа (*mtime*) изменилось.

Расширение LibreOffice для Cockpit

Введение

Расширение `LibreOffice` для Cockpit позволяет отключить использование макросов.

Установка расширения для Cockpit

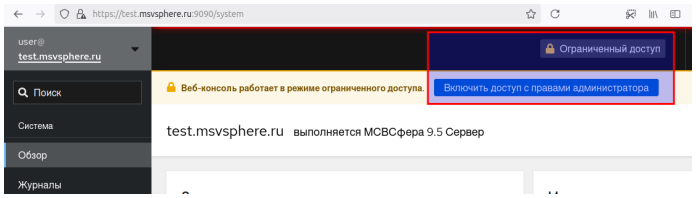
Установите соответствующее расширение для панели управления `Cockpit` с помощью следующей команды:

```
$ sudo dnf install cockpit-msvsphere-libreoffice
```

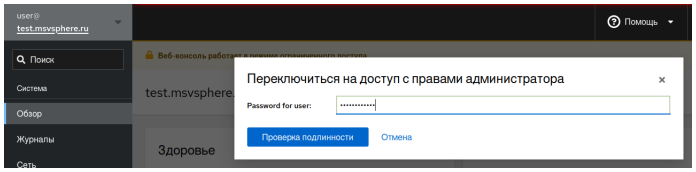
Настройка

Для отключения макросов требуются привилегии администратора:

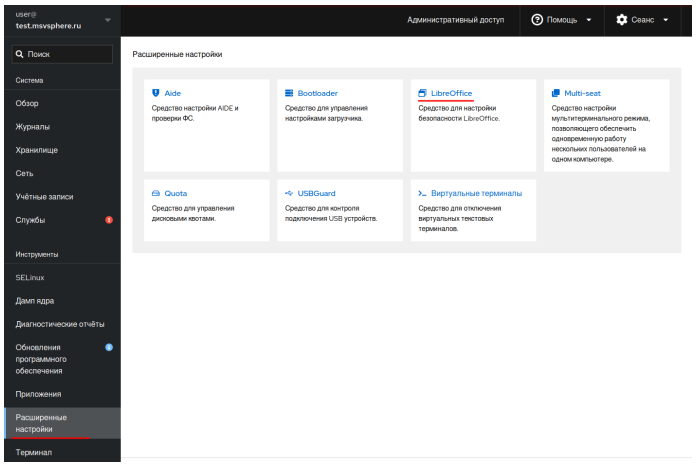
- войдите в панель управления `Cockpit`, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником.



- В открывшейся форме укажите свой пароль и нажмите на кнопку «Проверка подлинности».

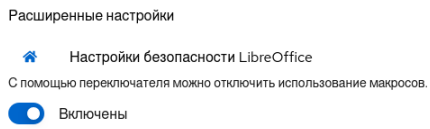


- После этого в левой панели откройте вкладку «Расширенные настройки», там перейдите по ссылке «LibreOffice»:

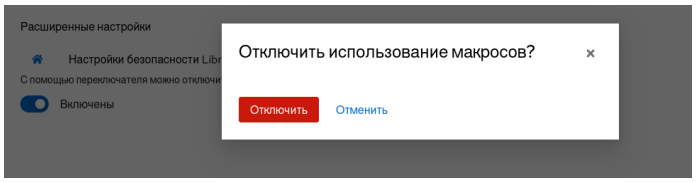


Отключение макросов

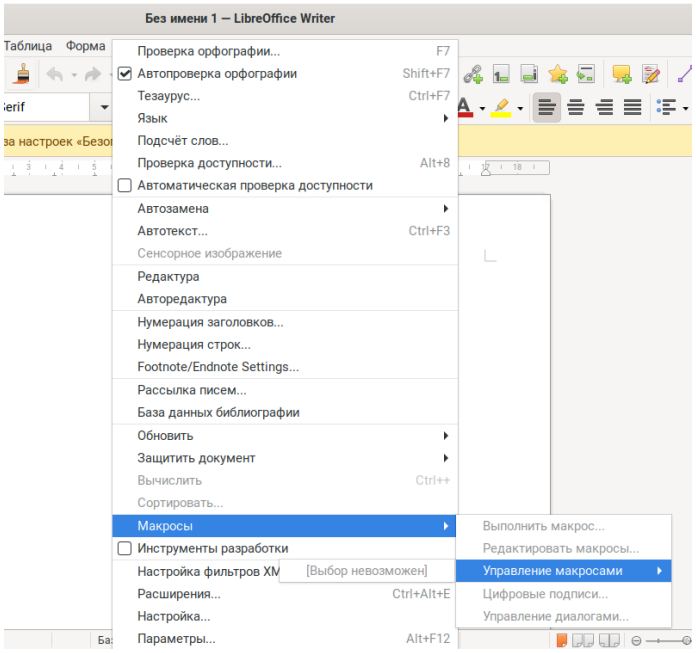
На странице отображается селектор, показывающий текущее состояние макросов (включены/отключены).

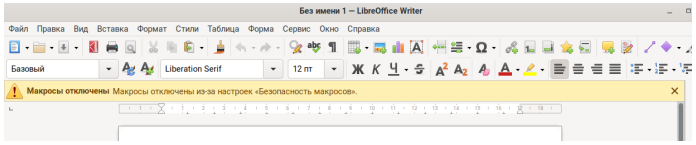


При отключении использования макросов будет показано окно подтверждения.



После подтверждения, использования макросов будет отключено.





Расширение Quota для Cockpit

Введение

Расширение `Quota` для Cockpit позволяет ограничить объем дискового пространства, используемого пользователями или группами, с помощью квот. Подсистема квот

управляет ограничениями на использование дискового пространства (`block`) и файлов (`inode`).

Установка расширения для Cockpit

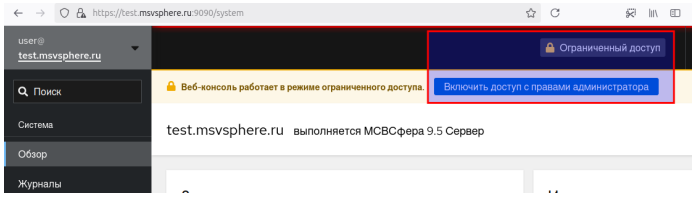
Установите соответствующее расширение для панели управления `Cockpit` с помощью следующей команды:

```
$ sudo dnf install cockpit-msvsphere-quota
```

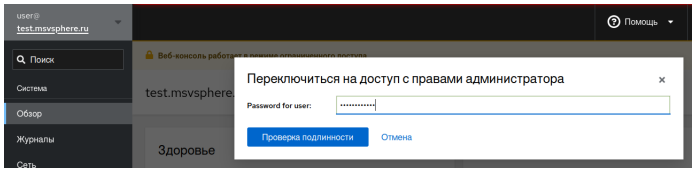
Настройка

Для управления квотами требуются привилегии администратора:

- войдите в панель управления `Cockpit`, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником.



- В открывшейся форме укажите свой пароль и нажмите на кнопку «Проверка подлинности».



- После этого в левой панели откройте вкладку «Расширенные настройки» (обозначена цифрой `1` на снимке экрана), там перейдите по ссылке «Quota» (обозначена цифрой `2` на снимке экрана):

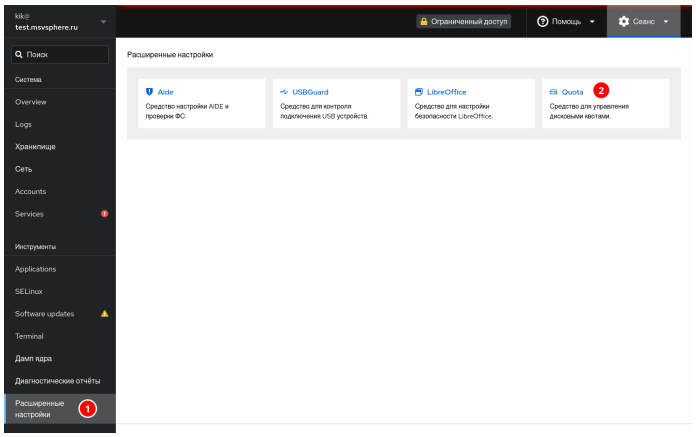
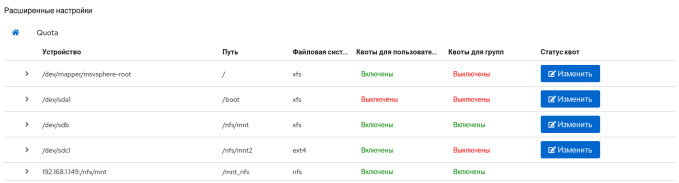


Таблица файловых систем

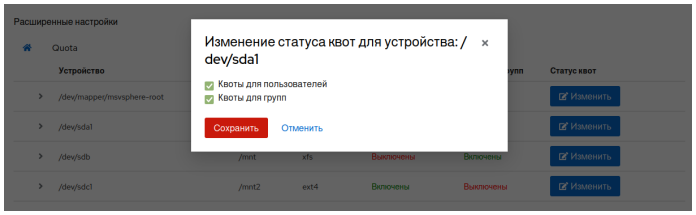
На странице отображаются поддерживаемые файловые системы.

- Устройство — дисковое устройство.
- Путь — путь к каталогу монтирования.
- Файловая система.
- Статус квот для пользователя — включены/выключены.
- Статус квот для групп — включены/выключены.



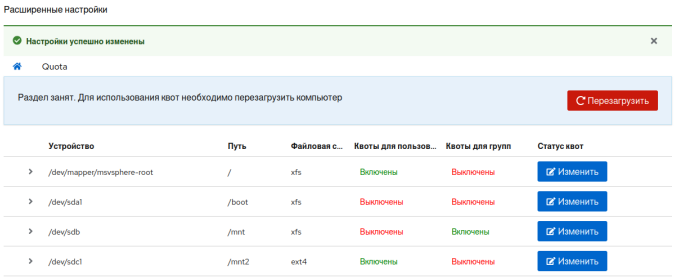
Изменение статуса квот

Для включения или выключения квот для пользователей/групп нажмите на кнопку «Изменить» для выбранной файловой системы. В открывшемся окне укажите нужный статус с помощью селекторов и сохраните изменения.



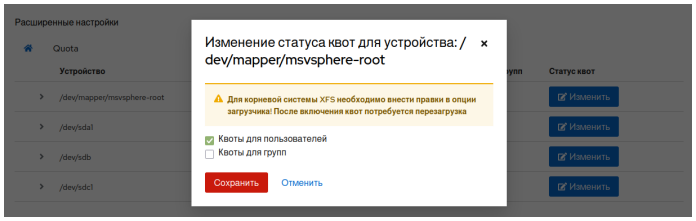
После сохранения, опции для включения квот будут добавлены в файл */etc/fstab*.

Также будет выполнено перемонтирование раздела. Если раздел занят и используется, для применения изменений необходимо перезагрузить систему.



Важно!

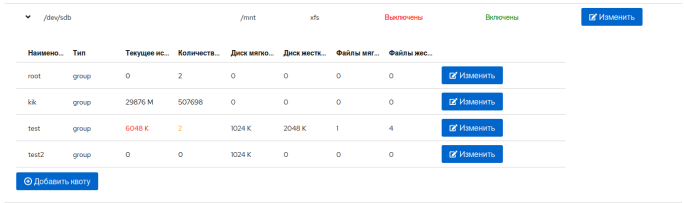
- Для включения квот корневой файловой системы `XFS` необходимо внести правки в опции загрузчика. В случае включения квот для корневой системы, будут добавлены опции для всех установленных ядер.



- Для файловой системы `NFS` нет возможности изменить статус квот. Включение/отключение квот для пользователей или групп нужно выполнять на сервере `NFS`.

Настройка квот пользователей/групп

Для просмотра текущих настроек квот для пользователей/групп нажмите на строку файловой системы, откроется таблица с данными. В зависимости от статуса квот будут отображены только пользователи/группы или все вместе.

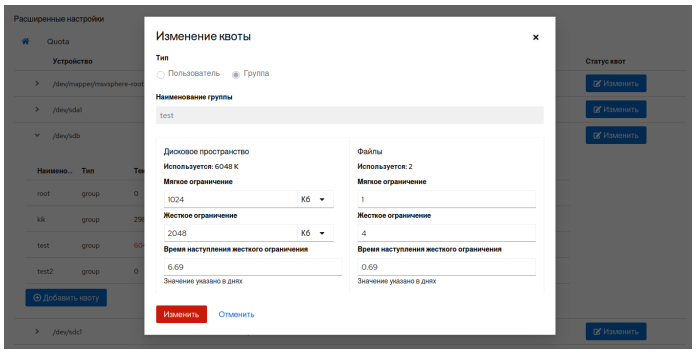


Список квот включает следующие данные:

- Наименование — имя пользователя/группы.
- Тип — пользователь или группа.

- Текущее использование дискового пространства.
- Текущее количество файлов.
- Мягкое ограничение дискового пространства — максимальный объём дискового пространства, доступный пользователю/группе. При его превышении отправляется уведомление, а после включения жёсткого ограничения (по умолчанию через 7 дней), мягкая квота становится жёсткой.
- Жёсткое ограничение дискового пространства — жёсткая квота на объём дискового пространства (при её превышении дальнейшая запись будет запрещена).
- Мягкое ограничение количества файлов — мягкая квота на количество файлов (при её превышении пользователю отправляется уведомление).
- Жёсткое ограничение количества файлов — жёсткая квота на количество файлов (при её превышении дальнейшая запись запрещена).

Для изменения квоты нажмите на кнопку „Изменить» для выбранного пользователя/группы.



Ограничение по объёму можно указать в килобайтах, мегабайтах, гигабайтах или терабайтах. Ограничение по файлам — в единицах. Если для пользователя/группы начался отсчёт времени до наступления жёсткого ограничения, то это значение будет также указано (в днях) и его можно изменить.

Важно!

Для файловой системы `NFS` нет возможности изменить время наступления жёсткого ограничения. Чтобы изменить значение, измените настройки квот на сервере `NFS` для связанной файловой системы.

Если существующий пользователь/группа не указаны в списке, то можно добавить квоту самостоятельно, нажав на кнопку «Добавить квоту».

Расширение «Виртуальные терминалы» для Cockpit

Введение

Расширение «Виртуальные терминалы» позволяет отключить использование виртуальных текстовых терминалов (переключение между ними осуществляется по нажатию клавиш `Ctrl+Alt+F3` — `Ctrl+Alt+F6`).

Установка расширения для Cockpit

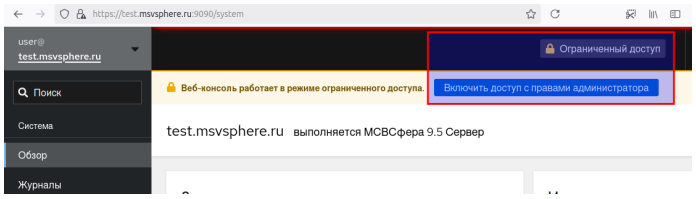
Установите соответствующее расширение для панели управления `Cockpit` с помощью следующей команды:

```
$ sudo dnf install cockpit-msvsphere-virt-terminal
```

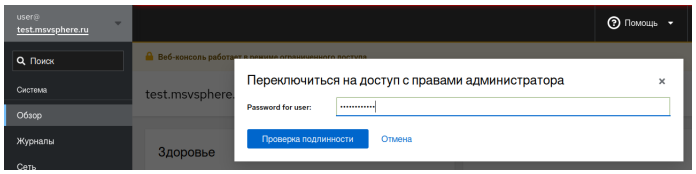
Настройка

Для отключения использование виртуальных терминалов требуются привилегии администратора.

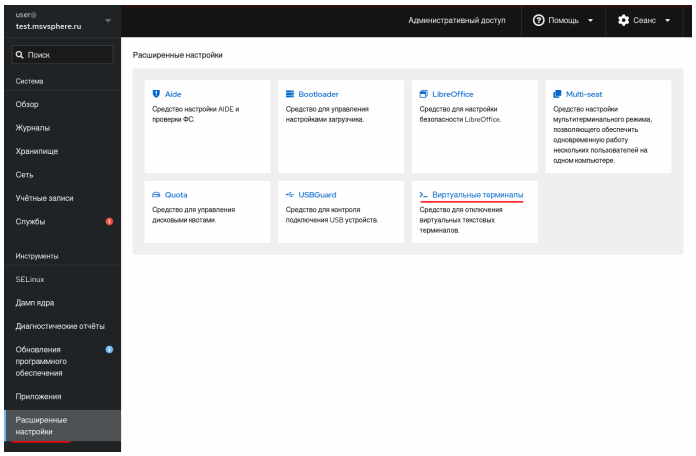
- войдите в панель управления `Cockpit`, на странице «Обзор» нажмите кнопку «Включить доступ с правами администратора» или кнопку «Ограниченный доступ», которая также доступна на других страницах системы управления. На приведённом ниже снимке экрана эти кнопки обозначены красным прямоугольником:



- В открывшейся форме укажите свой пароль и нажмите на кнопку «Проверка подлинности».

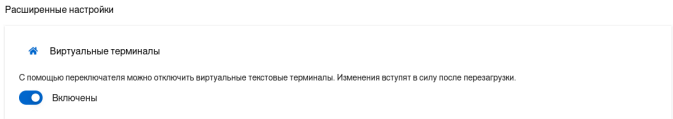


- После этого в левой панели откройте вкладку «Расширенные настройки», там перейдите по ссылке «Виртуальные терминалы».

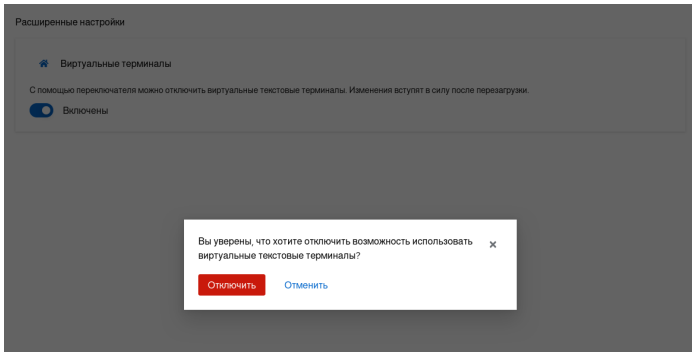


Отключение виртуальных текстовых терминалов

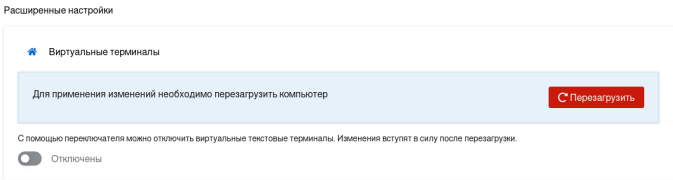
На странице отображается селектор, показывающий текущее состояние виртуальных терминалов (включены/отключены).



При отключении виртуальных терминалов будет показано окно подтверждения.



Настройка будет применена после перезагрузки. После подтверждения будет показана кнопка «Перезагрузить» (перезагрузка будет выполнена сразу после нажатия).



Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм	измененных	замененных	новых	аннулированных					