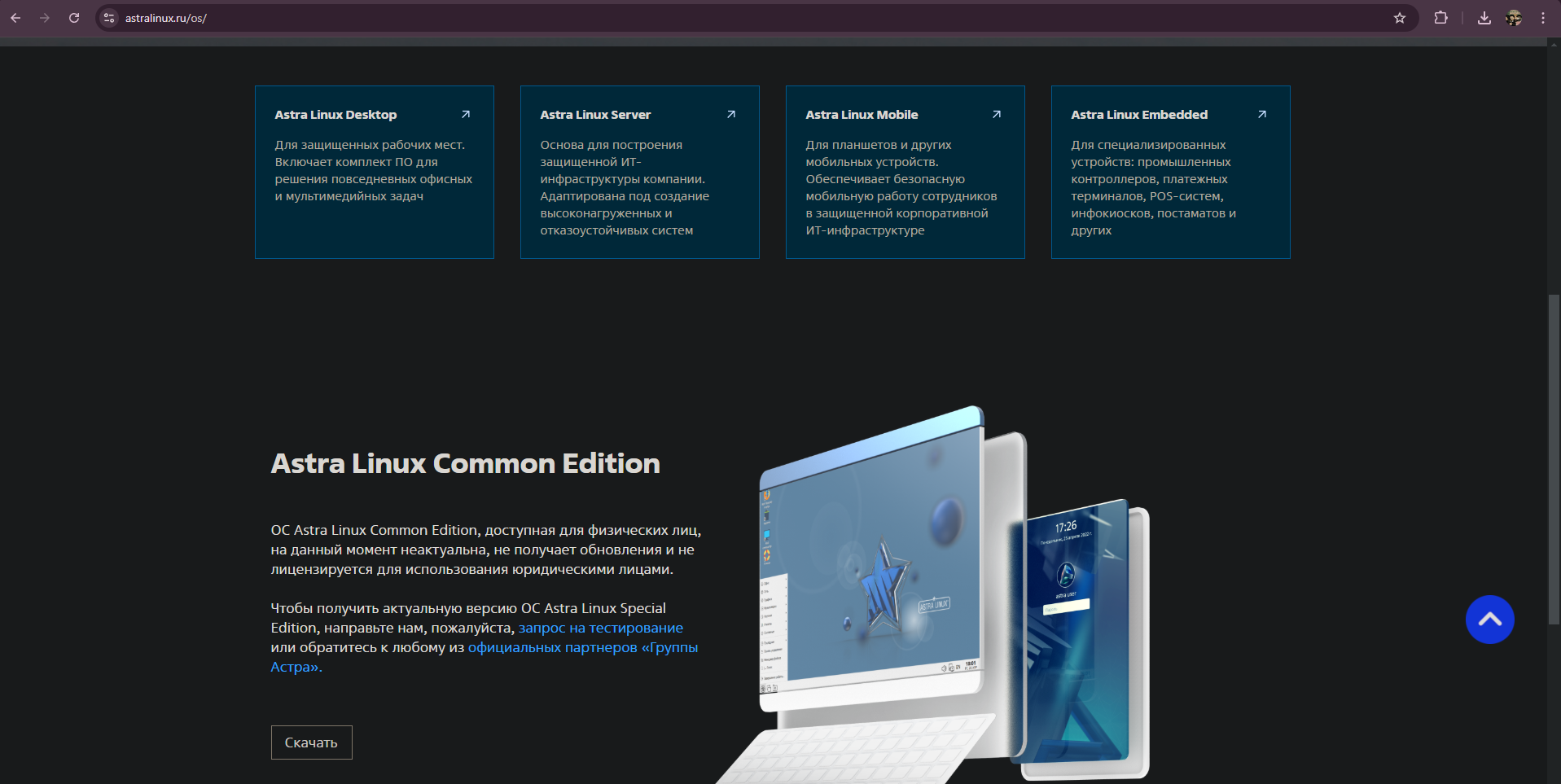
**1 Установка ОС**

1.1 Скачиваем дистрибутив Astra Linux с официального сайта:

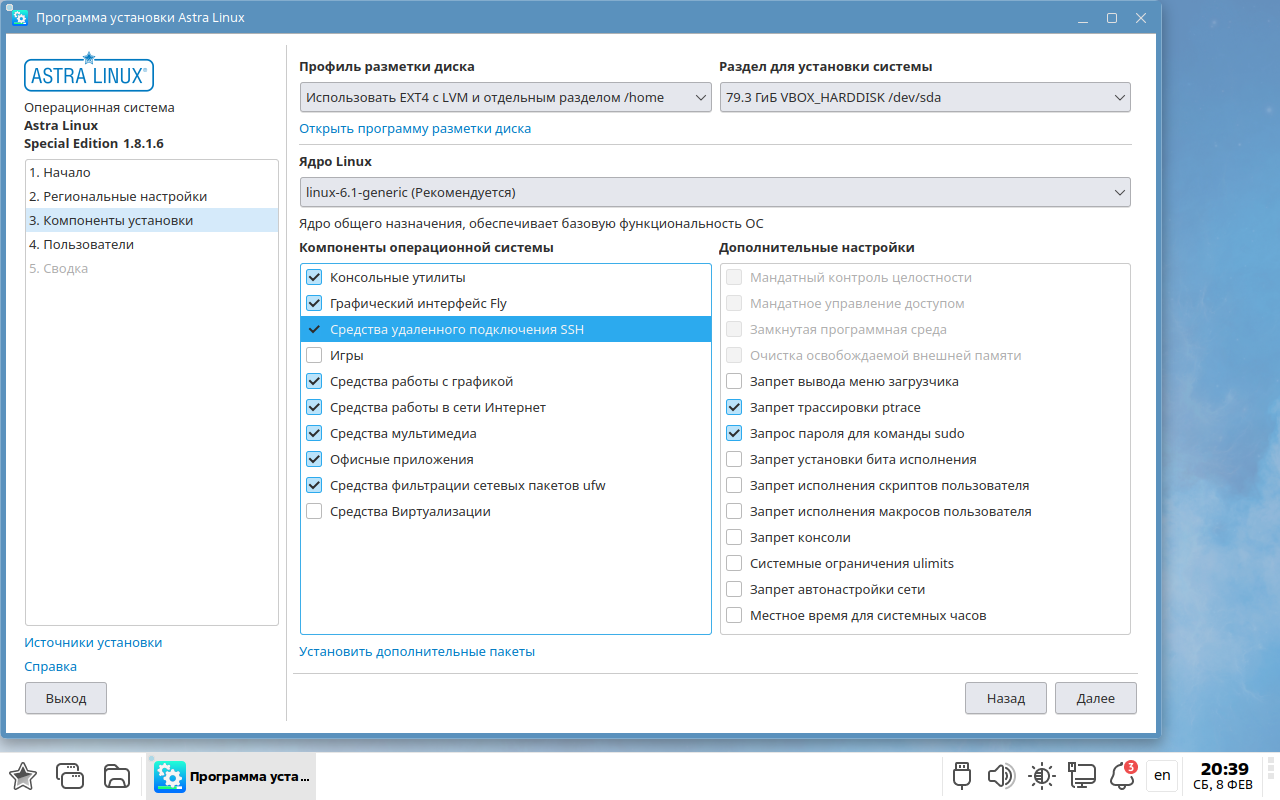


1.2 Устанавливаем данную ОС, присваиваем имя машины “sr-sazonov”:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

1.3 При настройке жесткого диска воспользуемся оснасткой «Авто» и при установке дополнительных компонентов указываем OpenSSH:



1.4 В поля «имя пользователя» вписываем фамилию Sazonov и указываем пароль:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

1.5 Настраиваем сетевой интерфейс, вводим команду sudo nano /etc/network/interfaces:



Изображение выглядит как текст, программное обеспечение, снимок экрана, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

Проводим настройку:

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

1.6 После настройки сетевых интерфейсов выполняем: sudo apt update, sudo apt upgrade для получения новых списков пакетов и обновление ОС:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1.7 Проверяем работоспособность системы:

Убеждаемся, что сеть настроена правильно:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Проверяем доступность SSH:

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**2 Задания и вопросы**

**2.1 Опишем процесс установки дистрибутива Asra Linux.**

2.1.1 Процесс установки:

Установка Astra Linux выполняется через графический интерфейс или терминал. Выбираем язык, раскладку клавиатуры, имя машины, имя пользователя и пароль. Затем настраиваем жесткий диск (вручную или автоматически) и устанавливаем дополнительные компоненты, например, OpenSSH (опционально).

2.1.2 Настройка сетевых интерфейсов:

* Статические записи: Редактирование файла /etc/network/interfaces
* Получение IP по DHCP: В файле /etc/network/interfaces указать iface eth0 inet dhcp.

2.1.3 Обновление ОС: используем команды sudo apt update и sudo apt upgrade.

2.1.4 Установка пакетов: используем команды sudo apt install <имя\_пакета>

**2.2 Изучим команду adduser**

Создаём тестовые учётную запись student:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Создаём тестовые учётную запись auditor:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

Разделение учетных записей позволяет ограничить доступ к системе и контролировать действия пользователей.

**2.3 Изучим команды groupadd, groupmod, usermod**

Создаём группы students и staff и добавляем в них созданных в предыдущем пункте пользователей соответственно:

Изображение выглядит как текст, снимок экрана, Шрифт

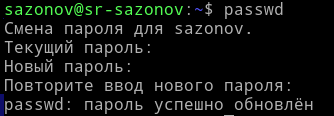
Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

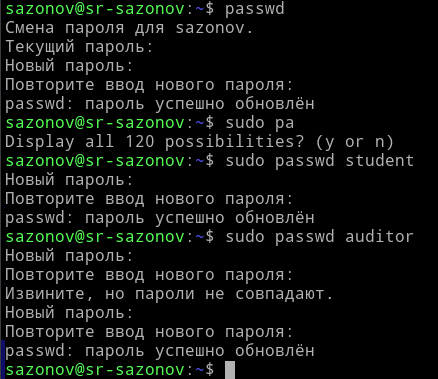
Группы позволяют организовать управление правами доступа, что снижает риски несанкционированного доступа.

**2.4 Изучим утилиту passwd**:

Сменим себе пароль:



Сменим пароль созданным учётным записям:



Применение для обеспечения безопасности:

Регулярная смена паролей и применение сложных паролей существенно повышают защиту учетных записей от несанкционированного доступа.

**2.5 Устанавливаем mc**:

2.5.1 Устанавливаем и запускаем mc:

Изображение выглядит как текст, снимок экрана, Шрифт

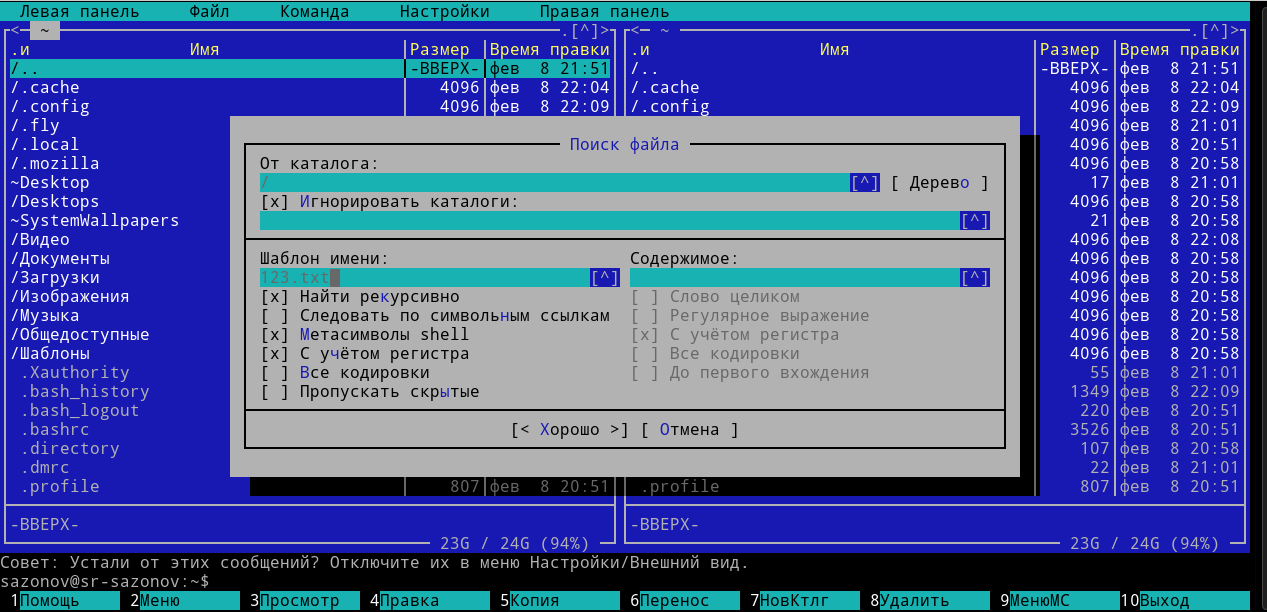
Контент, сгенерированный ИИ, может содержать ошибки.

2.5.2 Изучим функционал:

механизм поиска:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.



Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

создания файлов:

Изображение выглядит как текст, электроника, снимок экрана, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

редактирование:

Изображение выглядит как текст, электроника, снимок экрана, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дисплей, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

просмотр прав на файлы:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, дисплей, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

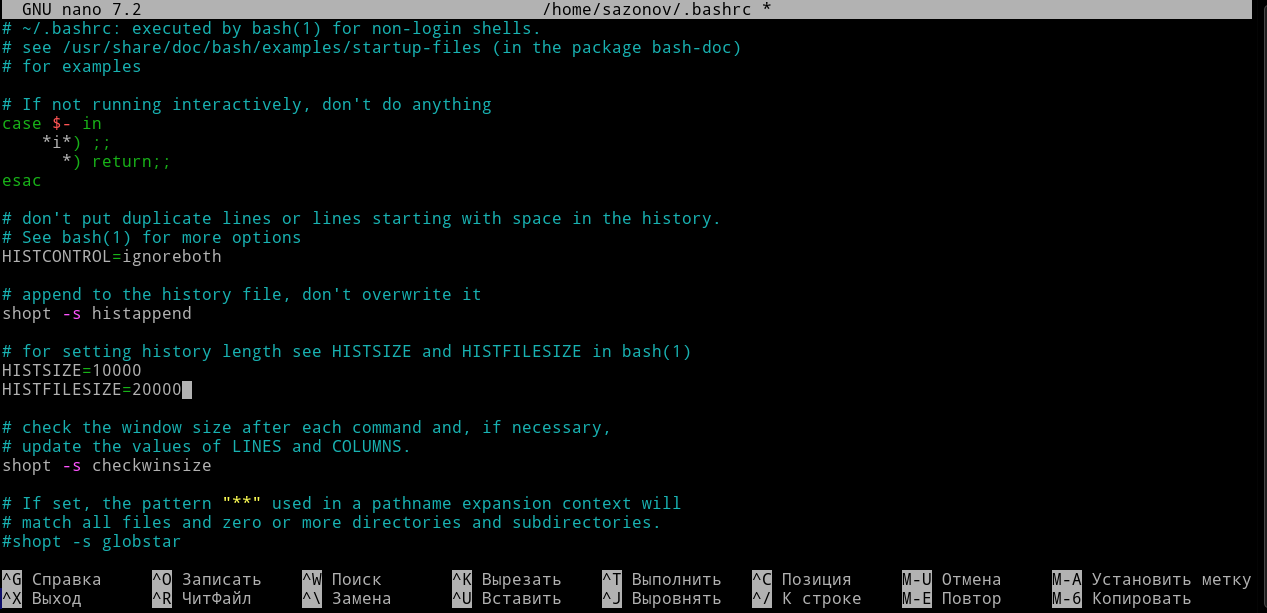
Инструмент mc упрощает управление файлами, помогает быстро проверить права доступа к файлам, обнаруживать неверно настроенные файлы и управлять ими.

**2.6 Изучим команду history**:

Увеличим размер истории:



Изменяем значения строчек HISTSIZE и HISTFILESIZE



Применение для обеспечения безопасности:

Расширенная история команд помогает в аудите действий пользователей, что важно для расследования инцидентов и контроля за выполненными командами.

**2.7** **Изучим команды grep, cat, ls.** (ключи, применение):

Команда grep ищет текстовые строки в файлах:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

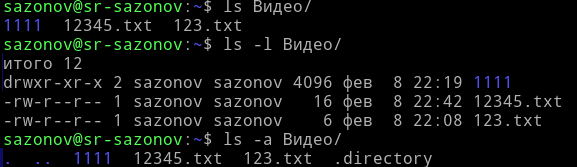
Команда cat выводит содержимое файла на экран:



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Команда ls выводит список файлов и директорий:



Применение для обеспечения безопасности:

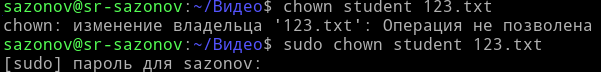
Эти команды позволяют анализировать журналы, проверять конфигурационные файлы и права доступа, что может помочь в обнаружении аномалий и нарушений безопасности.

**2.8 Изучим chmod, chown, chgrp** (продемонстрируем применение):

Chmod изменяет прав доступа:



chown изменяет владельца и группы файла:



chgrp изменяет только группы файла:

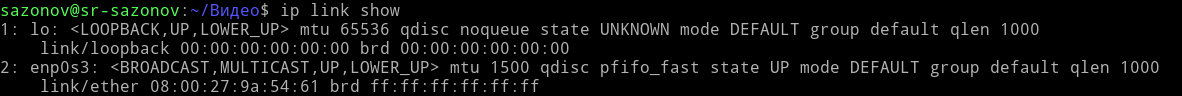


Применение для обеспечения безопасности:

Правильное управление правами доступа предотвращает несанкционированное чтение или изменение критичных файлов, снижая риск компрометации системы.

**2.9 Изучим утилиту ip**:

Просмотр интерфейсов:

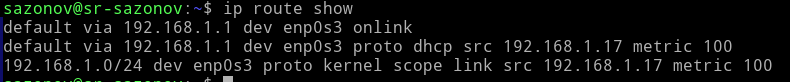


Мониторинг состояния сети:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Просмотр таблицы маршрутизации:

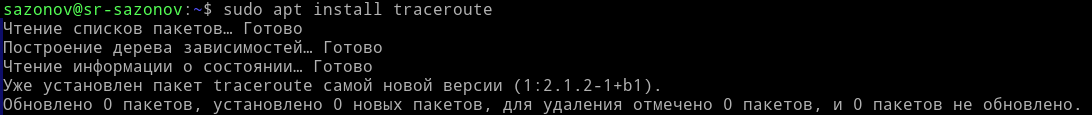


Применение для обеспечения безопасности:

Утилита ip позволяет детально отслеживать состояние сети, обнаруживать аномалии в конфигурации интерфейсов и маршрутизации, что критически важно для выявления атак и неправильной настройки сети.

**2.10 Изучим утилиту traceroute**:

Устанавливаем traceroute:



Применяем команду:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

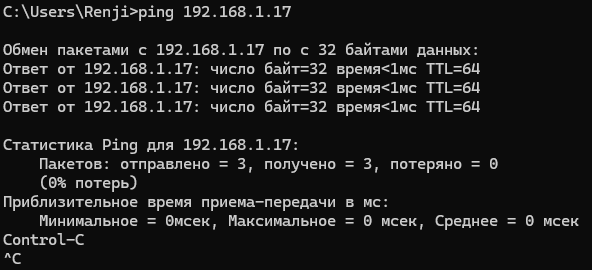
Утилита traceroute помогает выявить узлы, через которые проходит трафик.

**2.11 Изучим утилиту iptables** (заблокируем возможность применения утилиты ping извне)

Заблокируем возможность применения утилиты ping извне:



До применения команды:



После применения команды:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

Ограничение ICMP-пакетов помогает уменьшить возможности для сканирования сети злоумышленниками.

2.12 Настраиваем подключение к виртуальной машине по ssh, используя пароль; продемонстрируйте возможность подключения. Далее, настройте подключение только по ключу (заблокируйте возможность подключения по паролю); продемонстрируйте подключение.

2.12.1 Подключаемся по SSH:

Изображение выглядит как текст, снимок экрана, Шрифт

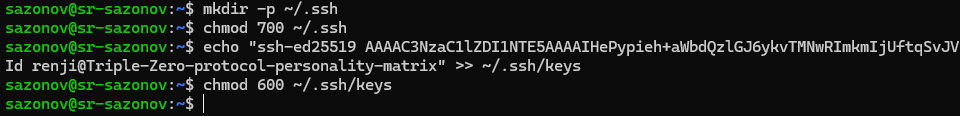
Контент, сгенерированный ИИ, может содержать ошибки.

2.12.2 Настройка подключения по ключу:

Генерация ключей на клиенте и передача публичного ключа на сервер:

Изображение выглядит как текст, снимок экрана, черный, черно-белый

Контент, сгенерированный ИИ, может содержать ошибки.



Теперь есть возможность подключаться, используя ssh ключ.

Отключаем аутентификации по паролю:



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



После изменения настроек:

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Пробуем подключиться по ssh ключу:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Применение для обеспечения безопасности:

Переход на аутентификацию по ключу значительно повышает безопасность, так как ключи сложнее подобрать или перехватить по сравнению с паролями.

2.13 Работа с журналами Linux: изучите инструменты journalctl и logwatch.

утилита Journalctl используется для просмотра системных журналов (логов):

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Создаём несколько подозрительных событий, например многократный неудачный вход в систему:

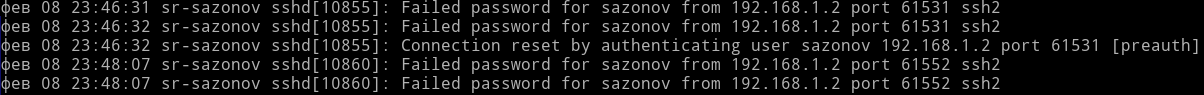
  
Изображение выглядит как текст, Шрифт, снимок экрана, черный

Контент, сгенерированный ИИ, может содержать ошибки.

Фильтруем по SSH:



Обнаружим данные события:



Применение для обеспечения безопасности:

Регулярный анализ логов с помощью journalctl позволяет обнаружить аномальные действия, своевременно отреагировать на попытки несанкционированного доступа и другие подозрительные события. Также логи дают возможность провести детальный аудит действий в системе, что важно для последующего анализа инцидентов безопасности.