МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра «Сети связи и системы коммутации»

Сборник лабораторных работ

По дисциплине:

«Основы работы с UNIX-подобными операционными системами»

Москва, 2024 г.

Содержание

Предисловие

В данном сборнике представлены основы работы с UNIX-подобными операционными системами, которые базируются на ядре Linux. Основная цель пособия — помочь обучающимся освоить базовые и промежуточные навыки системного администрирования, необходимые для управления операционной системой, работы с пользователями, процессами и сетью.

Методический материал ориентирован на использование современных дистрибутивов Linux, таких как Ubuntu. Для выполнения лабораторных работ рекомендуется использовать виртуальные машины или удалённые серверы, что позволяет безопасно экспериментировать с настройками системы.

Сборник дополнен контрольными вопросами и заданиями, направленными на закрепление изученного материала.

Вместе с этим мы акцентируем внимание на безопасности, которая является неотъемлемой частью работы системного администратора. Во всех заданиях учитываются базовые принципы безопасной работы: минимизация прав доступа, защита данных и использование современных подходов к настройке системы.

Введение

Для начала работы необходимо выбрать Linux-дистрибутив. Для ознакомления и выполнения лабораторных работ рекомендуется использовать дистрибутив Ubuntu (https://ubuntu.com/download/desktop), но обучающийся вправе выбрать любой иной. (Далее под словом «Linux» будет иметься в виду не ядро, а вся система в целом.)

Также обучающийся может установить любой Linux в качестве первой/второй системы, или подключиться по SSH к удалённому серверу, где уже установлена система

В ходе выполнения лабораторных работ рекомендуется использовать виртуальную машину. На компьютере у обучающихся должны быть включена возможность виртуализации в UEFI/BIOS. Затем необходимо выбрать средство виртуализации системы, где рекомендуется выбрать Oracle VM VirtualBox (https://www.virtualbox.org/wiki/Downloads), но обучающейся может выбрать иное ПО.

После установки VirtualBox или другого ПО, при осуществлении установки системы необходимо отконфигурировать параметры виртуальной машины. Рекомендуемые требования для корректной работы (см.таб.1).

Таблица 1 – Рекомендуемые требования операционной системы на 2024 год.

Параметры	**Требования**
Оперативная память	2ГБ
Процессор	2 Ядра
Дисковое пространство	25ГБ

После запуска виртуальной машины необходимо провести стандартную установку системы: выбрать язык, раскладку, часовой пояс, разметку на диск, имя пользователя и пароль. Обновляем систему и перезагружаем ее.

Лабораторная работа №1. Работа с директориями.

Цель работы

Ознакомиться с устройством файловой системы в Linux. Изучить методологию работы с файловой системой: создание директорий, файлов и т.д.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

`mkdir`: Создание новой директории.

- Пример: `mkdir my_dir`. Эта команда создает директорию с именем `my_dir`.

`touch`: Создание пустого файла.

- Пример: `touch file` создаст пустой файл с именем `file`.

`cd`: Переход в другую директорию.

- Пример: `cd my_dir`. Эта команда позволит перейти пользователю в директорию `my_dir`. Также можно использовать команду `cd` без аргументов для перехода в домашнюю директорию пользователя.

`ls`: Вывод списка файлов и директорий в текущей директории.

- Флаг `-l`: подробный список с дополнительной информацией о файлах (права доступа, владелец, размер и дата изменения); Флаг `-a`: выводит скрытые файлы (начинающиеся с точки); Флаг `-h`: отображает размер файлов.

- Пример: `ls -la`.

nano: Редактирование текстовых файлов.

- Пример: `nano file` откроет файл с названием `file`, где его можно отредактировать. Чтобы сохранить файл, используйте сочетание клавиш `CTRL+O`, а для выхода из `nano` используйте сочетание клавиш `CTRL+C`.

cp: Копирование файлов.

- Пример: `cp /путь/к/исходному/файлу /путь/к/целевой/директории` скопирует файл из одной директории в другую.

rm: Удаление файла.

- Пример: `rm /путь/к/файлу` удалит файл.

mv: Перемещение или переименование файлов.

- Пример (перемещение): `mv /путь/к/исходному/файлу /путь/к/целевой/директории` переместит файл в другую директорию.

- Пример (переименование): `mv text.txt config.conf` переименует файл `text.txt` в `config.conf`

cat: Просмотр содержимого файла и его базовое редактирование.

- Пример: `cat text.txt` покажет содержимое файла. Чтобы изменить файл, напишите `cat >> text.txt`. После допишите новые строки в файл и воспользуйтесь сочетание клавиш `CTRL+D`. Также, можно объединить несколько файлов: `cat file_1 file_2 file_3 > file_all` создаст новый

файл `file_all`, который содержит все строки из файлов `file_1 file_2 file_3`.

`head` и **`tail`**: Вывод первых или последних строк файла.

- Флаг `-n` (количество строк): определяет количество строк для вывода.

- Пример (первые 8 строк): `head -n 8 config.conf`.

- Пример (последние 3 строки): `tail -n 3 config.conf`.

`wc`: Подсчет символов, строк и байтов в файле.

- Флаг `-m` (characters): подсчитывает количество символов.

- Пример: `wc -m text.txt`

- Флаг `-l` (lines): подсчитывает количество строк.

- Пример: `wc -l text.txt`

- Флаг `-c` (bytes): подсчитывает размер файла в байтах.

- Пример: `wc -c text.txt`

Также, стоит дополнить, что в текстовых документах размер одного символа равен 1 байту, так что при использовании команды `wc -m` и `wc -c` с такими файлами пользователь увидит одинаковый результат.

`df`: Анализ занимаемого места на диске.

- Пример: `df -h` покажет информацию о дисковом пространстве в читаемом виде (т. к. используется флаг `-h`).

pwd: Отображение текущего рабочего каталога.

- Пример: `pwd` в домашней категории выведет: `/home/Имя_Пользователя`.

ln: Создание ссылки.

- Пример: `ln -s test.txt soft_link` создаст символическую ссылку на test.txt с именем soft_link. (т.к. используется флаг -s).

- Пример: `ln test.txt hard_link` создаст жесткую ссылку на test.txt с именем hard_link.

Задание

1) Уточните в какой директории находится пользователь. Создайте новую директорию с произвольным названием в домашнем каталоге пользователя. Перейдите в неё и создайте в ней пустой файл. Проверьте то, что файл создался.

2) В данной директории создайте ещё один файл, но уже с определённым расширением (текстовый документ txt). Заполните его произвольным текстом с помощью текстового редактора nano или Vim. Проверьте то, что файл создался.

3) Вернитесь в домашнюю папку и создайте новую директорию. Скопируйте ранее созданный текстовый документ из предыдущей директории в новую. Проверьте то, что файл находиться в новой директории. Удалите файл из предыдущей директории. Проверьте то, что файл удален в старой директории.

4) Переместите файл в первую директорию. Проверьте, что файл был перемещён. Удалите вторую директорию. Проверьте то, что директория удалена.

5) Перейдите в первую директорию. Выведите содержимое текстового документа, созданного в задании 2. Переименуйте этот документ, присвойте имя text_1. Создайте ещё два текстовых документа с названиями text_2.txt и text_3.txt и произвольным содержимым. Проверьте то, что файлы были созданы. Объедините все файлы в один, присвойте ему имя text_all и выведите его содержимое.

6) Находясь в этой же директории, дополните текстовый документ text_all, используя cat. Допишите два любых слова. Выведите результат. Выведите первые две строки файла. Выведите три последние строки файла. Выведите общее количество символов в файле. Выведите размер файла в байтах.

7) Проанализируйте занимаемое место на диске операционной системой.

8) Создайте символическую ссылку на файл, созданного в задании 5, в той же директории. Проверьте, что ссылка создалась.

9) Создайте жесткую ссылку на файл, созданного в задании 5, в той же директории. Проверьте, что ссылка создалась.

Контрольные вопросы

1. Перечислите основные каталоги корневого раздела.
2. Как скопировать директорию? Как удалить директорию?
3. Как перебраться из директории на 1 уровень выше?

4. Какие виды ссылок бывают? Чем они отличаются?

5. Расшифруйте команды: `pwd`, `cd`, `mv`, `rm`.

Лабораторная работа №2. Операции с пользователями в системе.

Цель работы

Ознакомиться с пользователями и группами в Linux. Изучить методологию работы с пользователями.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

`who`: Отображение пользователей, работающих в системе, с дополнительной информацией.

- Пример: `who` выведет информацию о пользователях, которые подключены к системе, в том числе и об терминальных сессиях, через которые происходит подключение.

`adduser`: Создание пользователя.

- Пример: `adduser test` запустит процесс создания пользователя с именем test, включая присвоения пароля, полного имени и т.д.

- Пример: `adduser -u 1111 test` запустит процесс создания пользователя с именем test и UID 1111. (т.к. используется флаг -u).

- Пример: `adduser -g 1111 test` запустит процесс создания пользователя с именем test и GID 1111. (т.к. используется флаг -g).

`whoami`: Отображение пользователя, работающего в системе.

- Пример: `whoami` выведет действующий идентификатор пользователя.

`users`: Отображение пользователей, работающих в системе.

- Пример: `users` выведет список регистрационных имен пользователей, работающих в настоящий момент в системе, в компактной, однострочной формате.

`userdel`: Удаление пользователей.

- Пример: `userdel user` удалит пользователя user.

`groupadd`: Создание групп.

- Пример: `groupadd test` создаст группу test.

`passwd`: Создание пароля пользователю.

- Пример: `passwd` запустит процесс по смене пароля.

`usermod`: Изменение параметров пользователя.

- Пример: `usermod -g 1111 test` сменит пользователю test на группу с идентификатором 1111 (т.к. используется флаг -g).

`groupmod`: Изменение параметров группы.

- Пример: `groupmod -n test1 test` сменит название группы с test на test1.

`groups`: Просмотр групп.

- Пример: `groups user` выведет список групп, в которых состоит user.

`groupdel`: Удаление группы.

- Пример: `groupdel test` удалит группу test.

Задание

1) Определите какие пользователи находятся в системе. Определите под каким пользователем вы вошли в систему. Затем выведите и дополнительную информацию о пользователях.

2) Смените пароль у пользователя.

3) Создайте нового пользователя. Проверьте, что пользователь создался. Смените созданному пользователю пароль.

4) Создайте пользователя с определенным UID. Создайте пользователя в определенной директории. Проверьте, что пользователи создались.

5) Создайте группу. Смените созданной группе имя. Создайте пользователя с определенным GID, созданным ранее. Проверьте, что пользователь и группа создались.

6) Добавьте в группу пользователя, созданного в пункте 4. Проверьте, что пользователь добавился в группу.

7) Удалите всех созданных пользователей. Проверьте, что пользователи удалились. Удалите группу. Проверьте, что группа удалилась.

Контрольные вопросы

1. В чем разница useradd и adduser?
2. В чем разница who и w?
3. Может ли пользователь принадлежать к нескольким группам?
4. Как проверить, что пользователь создался?
5. Какая информация о пользователе содержится в системе?

Лабораторная работа №3. Изучение флагов в командах. Основы работы с различными правами доступа. Работа с процессами.

Цель работы

Ознакомиться с флагами и с различными правами доступа в Linux. Изучить методологию работы с пользователями с различными правами доступа и использование флагов в командах.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться. Далее в разделе краткой теории не будут упоминаться флаги в командах - используйте `--help`.

«test» `--help`: Отобразит краткую справочную информацию.

- Пример: `ls --help` выведет краткую справочную информацию о команде `ls`, объясняющую основные опции и функциональность команды.

`sudo`: Утилита, позволяющая пользователю выполнять команды с привилегиями суперпользователя или иного пользователя, не выходя из своей учетной записи.

`man «test»`: Отобразит руководство по команде «test».

- Пример: `man ls` выведет руководство о команде `ls`, объясняющую опции и функциональность команды в т.ч. возможные флаги и переменные.

chmod: Утилита, используемая для изменения прав доступа (разрешений) к файлам и директориям.

- Пример: `chmod a-w file.txt –` убирает право на запись для всех к файлу file.txt.

ps: Утилита, используемая для отображения информации о текущих процессах, запущенных в системе.

kill: Утилита, предназначенная для отправки сигналов процессам. Несмотря на название, **kill** не всегда используется для завершения процесса.

- Пример: `kill <PID>` PID – идентификатор процесса, который вы хотите завершить или которому хотите отправить сигнал.

top: Утилита отображает список процессов, их ресурсоемкость и состояние, а также общую информацию о загрузке процессора, памяти и других системных метрик

Задание

1) Воспользуйтесь командой `ps или top`, а затем завершите, перезапустите, приостановите и возобновите любой процесс, не относящийся к стандартным.

2) Создайте папку, в которой только ваш пользователь сможет создавать и удалять файлы, а остальные пользователи – только просматривать список файлов.

3) Создайте «теневой» каталог. Покажите, что получить список файлов нельзя, но можно получить доступ к файлу, зная его имя.

4) Создайте каталог, доступ на запись, в которую есть только у суперпользователя, а доступ на чтение – у всех. Попробуйте создать файл, а затем с правами суперпользователя.

5) Создайте файл, отредактировать который сможет только суперпользователь, а прочитать – любой.

6) Создайте нового пользователя. Создайте новую группу под названием test. Добавьте своего пользователя и нового пользователя в группу test (не делайте эту группу основной). Создайте в домашней директории нового пользователя файл (вы получите ошибку). Создайте новую папку /home/test. Назначьте владельцем группу test. Выдайте группе полные права на директорию. Убедитесь, что теперь оба пользователя в группе test могут создавать файлы в папке /home/test.

Контрольные вопросы

1. Разница между утилитами sudo и sudo su?
2. В чем разница между SUID, SGID и Sticky Bit?
3. Разница между утилитами top и ps?
4. Опишите, как можно снять все права на папку для всех, кроме владельца с помощью одной команды.
5. Как узнать, какой процесс занимает больше всего процессорного времени, и как его завершить?

Лабораторная работа №4. Утилиты.

Цель работы

Ознакомиться с различными утилитами в системе Linux, изучить их функциональность и способы работы, включая установку, настройку и использование для решения типичных задач. Изучить работу с пакетным менеджером, а также с утилитами для архивирования, мониторинга процессов, загрузки файлов и работы с сессиями терминала.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

`tar`: Утилита для создания архивов и распаковки файлов.

Установка утилит и обновление в UNIX-подобных ОС может отличаться друг от друга.

- Пример: `sudo apt install <название_утилиты>` - Здесь используется пакетный менеджер apt, работающий с репозиториями, прописанными в /etc/apt/sources.list.

- Пример: `sudo apt-get install <название_утилиты>` - обновит заданную утилиту до последней версии.

`htop`: Интерактивная утилита для мониторинга процессов.

Также, стоит рассмотреть то, **как удалять утилиты** правильно – со всеми конфигурационными файлами и прочими дополнениями.

- Пример: 1. `sudo apt purge <название_утилиты>; 2. sudo apt autoremove; 3.sudo apt clean` - Первая команда удалит утилиту. Вторая удалит ненужные зависимости. Третья очистит кэш загруженных пакетов.

`wget`: Утилита для загрузки файлов из сети, поддерживающую HTTP, HTTPS и FTP.

- Пример: `wget -O logo.svg https://upload.wikimedia.org/wikipedia/commons/7/76/Ubuntu-logo-2022.svg` - скачает логотип Ubuntu в формате SVG с сайта Wikipedia.

Иногда бывают такие ситуации, когда какая-либо специфическая утилита не находится в репозиториях, которые идут по умолчанию. Например, утилита Neofetch. Если вы попробуете скачать её просто так, без принудительного добавления дополнительных репозиториев, то вам выдастся ошибка.

Попробуем в качестве примера добавить репозиторий и скачать эту утилиту. Откроем файл /etc/apt/sources.list:

```
sudo nano /etc/apt/sources.list
```

Добавим в этот файл дополнительную строку:

```
deb http://ppa.launchpad.net/dawidd0811/neofetch/ubuntu focal main
```

Сохраним файл и выйдем из редактора. Добавим GPG-ключ для нового репозитория (Не всегда необходимы ключи):

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys EEA14886
```

Обновим репозиторий и установим утилиту:

```
sudo apt update
```

```
sudo apt install neofetch
```

Таким образом можно редактировать список репозиториев, чтобы устанавливать специфические пакеты и утилиты.

`tmux`: Утилита позволяет создавать множество сессий, которые будут работать на фоне. Спектр применения широкий, как, к примеру, работа сразу с двумя проектами. Также со всеми возможности можно ознакомиться https://tmuxcheatsheet.com/ на официальном сайте.

- Пример: `tmux kill-session -t <session_name> -` закроет сессию.

Задание

1) Создайте новую сессию tmux с названием system_monitoring. Разделите в ней терминал на четыре панели в виде сетки (два по горизонтали и два по вертикали). Покиньте сессию и создайте новую с любым названием.

2) Присоединитесь к сессии system_monitoring. Откройте htop в первой панели. Во второй панели воспользуйтесь утилитой wget и выполните нижеприведённые пункты:
   2.1) Скачайте с названием «primer» файл с иного сайта.
   2.2) Выполните рекурсивную загрузку.
   2.3) Скачайте файл с иного сайта и ограничьте его скорость загрузки.
   2.4) Увеличьте кол-во попыток загрузки файла.

3) В сессии system_monitoring. Во третьей панели воспользуйтесь утилитой tar и выполните нижеприведённые пункты:
   3.1) Создайте архив с названием «test» и заархивируйте произвольных 2 файла.
   3.2) Извлеките файлы из архива «test».
   3.3) Покажите содержимое архива без извлечения.
   3.4) Добавьте произвольный файл в архив «test».
   3.5) Создайте архив расширения tar.gz методом архивации Gzip2.
   3.6) Создайте архив расширения tar.bz методом архивации Bzip2.

4) Переключитесь с сессии system_monitoring на другую сессию, не выходя из неё. Попробуйте скачать архив с утилитой Dust по ссылке: https://github.com/bootandy/dust/releases/download/v1.1.1/dust-v1.1.1-x86_64-unknown-linux-gnu.tar.gz (для x86-x64 систем), https://github.com/bootandy/dust/releases/download/v1.1.1/dust-v1.1.1-arm-unknown-linux-gnueabihf.tar.gz (для ARM-систем) или https://github.com/bootandy/dust/releases/download/v1.1.1/dust-v1.1.1-aarch64-unknown-linux-gnu.tar.gz (для AArch64 систем). Если ссылки по неким причинам станут не доступны, то необходимо скачать на ваш выбор иной архив с утилитой. И выполнить ниже представленные пункты, с условностью вашей выбранной утилитой. Распакуйте архив. Перенесите исполняемый файл из этой папки в директорию с утилитами.

5) Вернитесь в предыдущую сессию system_monitoring. В четвёртой панели напишите команду dust. Данная утилита отобразит состояние диска в псевдографическом интерфейсе по аналогии с du.

6) Покиньте и завершите сессию. Удалите архивы и папки из домашней директории, созданные в течение выполнения задания.

Контрольные вопросы

1. Что такое пакетный менеджер?
2. Какие утилиты используются для работы с файлами и архивами?
3. Какие основные показатели отображает утилита htop? Как можно сортировать процессы в утилите?
4. Можно ли запускать утилиты не только через терминал?
5. Чем отличаются утилиты от обычных программ?

Лабораторная работа №5. Изучение файла настроек Shell, команды alias и переменных окружения.

Цель работы

Ознакомиться с файлами настроек Shell и их ролью в настройке пользовательской среды. Изучить команды для создания и управления alias, а также переменные окружения, их назначение и использование в системе Linux.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться. Alias (Псевдоним команды) — это команда, состоящая из одного слова, но выполняющая другую заданную команду со всем набором опций. Переменные окружения — это специальные переменные, определенные оболочкой и используемые программами во время выполнения. Они могут определяться системой и пользователем.

echo: Команда, которая используется для вывода текста или значений переменных в стандартный вывод.

- Пример: `echo $TEST` знак `$` перед именем переменной сообщает Shell, что нужно вывести ее значение.

Задание

1) Откройте файл «.bashrc» и дополните его своим собственным alias. Затем продемонстрируйте работу вашего alias.

2) С помощью переменной окружения выведите путь к домашней директории пользователя.

3) Смените в переменной, которой вы воспользовались в 2-ом пункте задания, иную домашнюю директорию.

4) Создайте и используйте собственную переменную окружения, задав ей любое значение. Выведите значение этой переменной.

Контрольные вопросы

1. В чем разница между файлами bashrc, .bash_profile и /etc/profile?
2. Опишите механизм работы переменных окружения при запуске дочерних процессов. Например, как передаются переменные от родительского Shell к запущенному скрипту?
3. Как настроить переменную окружения так, чтобы ее значение было доступно только для определенного пользователя, но и сохранялось между сессиями?
4. Какие риски связаны с добавлениями пользовательских переменных окружения или alias в общедоступные файлы настроек, такие как /etc/profile?
5. Что произойдет, если вы измените значение переменной PATH на некорректное? Как это может повлиять на работу системы?

Лабораторная работа №6. Скрипты и планирование выполнения команд.

Цель работы

Ознакомиться с основами написания и выполнения Shell-скриптов. Изучить методы автоматизации задач с помощью планировщика `cron`, его настройку и использование для периодического выполнения команд в системе Linux.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

`at`: Утилита предназначена для выполнения задач или команд в указанное время.

В Ubuntu Shell-скрипты можно выполнять различными способами. Каждый из них имеет особенности, связанные с контекстом, правами доступа и использованием интерпретатора. Рассмотрим их подробно:

1. Выполнение через явное указание интерпретатора

   Этот способ используется, когда необходимо указать, какой интерпретатор следует использовать для выполнения скрипта.

   - Пример: `bash script.sh`

2. Выполнение с помощью команды ./

   Этот метод используется для выполнения скриптов как самостоятельных программ, если у скрипта есть разрешение на выполнение.

   - Пример: `./script.sh`

3. Выполнение с использованием `sudo`

Если скрипт требует привилегий суперпользователя, его необходимо запускать с помощью `sudo`. Однако, метод выполнения через `sudo ./script.sh` может быть проблематичным из-за особенностей обработки переменных среды. Рекомендуется использовать явное указание интерпретатора.

- Пример: `sudo bash script.sh`

4. Выполнение через графические оболочки или планировщики

Скрипты могут быть выполнены через другие средства, такие как планировщики (например, `cron.sh`) или графические оболочки. В таких случаях важно указать полный путь к скрипту и интерпретатору.

Задание

1) Напишите скрипт, который:
   1.1) Архивирует содержимое домашней директории с помощью `tar`, предварительно заполните домашнюю директорию любыми файлами.
   1.2) Добавляет текущую дату в имя архива (например, backup_2024-10-09.tar.gz).
   1.3) Перемещает в архив в каталог резервный копий (например, в /home/user/test.)
   1.4) Настройте задание в `cron`, чтобы скрипт выполнялся ежедневно в 02:00.

2) Напишите скрипт, который:
   2.1) Находит и удаляет временные файлы (например, файлы с расширением .tmp или .log, старше 7 дней) в директории загрузок.

2.2) Настройте выполнение задания через `cron` раз в день.

3) Напишите скрипт, который:
   3.1) Определяет время работы системы с момента последней перезагрузки.
   3.2) Если система работает более 30 дней без перезагрузки, то перезагружает.
   3.3) Настройте выполнение через `cron.`

4) Напишите скрипт, который:
   4.1) Проверит доступное место на диске.
   4.2) Если свободное место меньше 20%, то выведет уведомление об этом в терминал.

Контрольные вопросы

1. В чем разница между использованием двойных и одиночных кавычек при задании строк?
2. Как запланировать выполнение скрипта каждые 15 минут, но только в рабочие дни?
3. Объясните, почему команды, работающие с пользовательским окружением, могут не работать через `cron`?
4. Как в скрипте реализовать обработку ошибок выполнения команды?
5. Какие риски связаны с автоматической перезагрузкой системы через скрипт?

Лабораторная работа №7. Изучение команд для настройки сети.

Цель работы

Ознакомиться с базовыми настройками сети. Изучить использование протокола SSH для безопасного удалённого доступа к UNIX-подобным ОС, включая настройку аутентификации с помощью ключей.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

Базовая настройка сети в UNIX-подобных ОС подразумевает под собой настройку IP-адресов, DNS, шлюзов и проверку состояния сети.

`ping`: Проверяет доступность узла по сети.

`ip a`: Просмотр текущих настроек сети: Данная команда покажет текущие интерфейсы подключения, IP-адреса, статус и дополнительные параметры.

`1. sudo ip addr add 192.168.1.100/24 dev eth0`: Первая команда назначает статический адрес интерфейсу eth0 (FastEthernet 0/0).

`2. sudo ip link set eth0 up`: Вторая команда активирует интерфейс.

`3. sudo ip route add default via 192.168.1.1`: Третья команда производит настройку шлюза.

Настройка DNS-серверов происходит в конфигурационном файле по пути /etc/resolv.conf, который можно открыть любым текстовым редактором. Например, можно добавить DNS-сервера от Google:

```
nameserver 8.8.4.4

nameserver 8.8.8.8
```

Проверить подключение можно командой ping:

```
ping -c 4 google.com

ping -c 4 192.168.1.1
```

SSH (Secure Shell) – сетевой протокол, который предназначен для безопасного удалённого доступа к UNIX-подобным ОС. Данный протокол используется повсеместно, например для доступа системного администратора к серверам, находящийся в центре обработки данных.

Этот сетевой протокол по умолчанию работает на порту 22. Но для обеспечения дополнительной безопасности его можно изменить в конфигурационном файле.

Основной конфигурационный файл SSH находится по пути /etc/.ssh/ssh_config. Рассмотрим ключевые параметры в этом файле (все параметры указаны по умолчанию для Ubuntu):

1. port 22. Порт, по которому идёт подключение;
2. PermitRootLogin yes. Указывает, можно ли подключиться по SSH от пользователя root.

3. PasswordAuthentication yes и UsePAM yes. Указывает, можно ли подключиться по SSH с помощью пароля от какого-либо пользователя. В противном случае для подключения может использовать только ключ.

Для подключения по SSH используется команда ssh. Пример команды для подключения к машине по адресу 132.212.19.9:

`ssh root@132.212.19.9 root_password`. Здесь указывается имя пользователя, с которого идёт подключение, IP-адрес сервера (или домен) и пароль пользователя.

Главная проблема – то, что при повторных подключениях придётся писать пароль заново. Для того, чтобы сразу подключатся нужному серверу, можно создать локальную конфигурацию. В UNIX-подобных ОС её необходимо создать по данному пути: ~/ssh/config. В Windows: C:\Users\username\.ssh\config.

Рассмотрим пример локальной конфигурации для нашего некого сервера:

```
host test_server
        hostName 132.212.19.9
        user root
        port 22
        identityFile \Users\username\test_server_key
```

Заметим, что в локальных конфигурациях отсутствует строка с паролем. Для подключения может использоваться только пара ключей. Создать ключ необходимо на локальном компьютере этой командой (для UNIX-подобных ОС и для Windows данная команда одинакова):

```
ssh-keygen -t rsa -b 4096 -f \Users\username\keys\test_server_key
```

После её введения создастся пара ключей: test_server_key и test_server_key.pub – приватный и публичный ключ соответственно. Приватный ключ предназначен для локальной машины, а публичный – для удалённого сервера. Чтобы перенести публичный ключ на удалённый сервер, используется данная команда:

```
ssh-copy-id root@132.212.19.9
```

После введения данной команды на удалённом сервере будет сохранён публичный ключ, и теперь с локального компьютера можно подключиться к нему без написания пароля:

`ssh test_server.` `test_server` в данном случае – название сервера, который был написан в локальной конфигурации.

После создания пары ключей можно изменить основной конфигурационный файл SSH на удалённом сервере и изменить параметр PasswordAuthentication и UsePAM на no. Таким образом подключение может осуществляться только через ключи.

Задание

1) Попробуйте добавить сторонние DNS-сервера. Удостоверьтесь, что сеть работает командой ping.
2) Проведите диагностику сети и продемонстрируйте таблицу маршрутов.
3) Попробуйте подключиться по SSH к Вашей виртуальной машине и/или Вашему компьютеру под управлением UNIX-подобной ОС с другого ПК.
4) Создайте пару ключей для подключения по SSH. Измените порт на 55555. Подключитесь заново.

5) Создайте нового пользователя. Создайте в папке пользователя директорию /.ssh. Дайте права -rwx----- (chmod 700) этой директории. Создайте внутри этой директории файл authorized_keys. Дайте права -rw------ (chmod 600) этому файлу. Скопируйте содержимое файла публичного ключа с локального компьютера в файл authorized_keys (командой echo “содержимое” >> /home/username/.ssh/authorized_keys). Дайте права r пользователю на чтение этого файла (chown -R username:username /home/username/.ssh). Попробуйте подключиться с локального ПК к виртуальной машине.

Контрольные вопросы

1. На каком уровне модели OSI и TCP/IP работает протокол SSH?

2. Какой порт может использоваться для подключения по SSH?

3. Можно ли подключаться по SSH с помощью локального конфигурационного файла с использованием пароля?

4. За что отвечает публичный ключ, а за что локальный? Каковы риски при утечке этих файлов?

5. Можно ли по SSH передать файл с помощью обычной команды cat?

Лабораторная работа №8. Анализ сетевого трафика и изучение инструментов для перенаправления результатов работы команд.

Цель работы

Ознакомиться с методами анализа сетевого трафика с использованием инструментов командной строки. Изучить возможности перенаправления ввода, вывода и ошибок в Shell для автоматизации работы с командами и управления результатами их выполнения.

Краткая теория

Все нижеприведенные команды справедливы для Ubuntu. В зависимости от дистрибутива команды могут видоизменяться.

`tcpdump`: Основной инструмент для захвата и анализа сетевых пакетов.

`netstat`: Используется для просмотра текущих сетевых подключений, открытых портов и маршрутов.

`traceroute`: Показывает маршрут прохождения пакетов до указанного адреса.

Перенаправление ввода и вывода

В Linux используется стандартная система потоков данных:

Стандартный ввод (stdin) — поток ввода (обычно клавиатура).

Стандартный вывод (stdout) — вывод успешных сообщений в терминал.

Стандартный поток ошибок (stderr) — вывод сообщений об ошибках.

Пайпы используются для передачи результата одной команды в другую. Примеры:

Передача списка файлов, содержащих "log", в подсчёт строк:

```
ls | grep log | wc -l
```

Поиск строки "error" в системном логе:

```
cat /var/log/syslog | grep error
```

Комбинирование ввода, вывода и сетевых утилит:

- Пример: `ping -c 4 google.com > ping_results.txt`

Задание

1) Анализ сетевого трафика:

1.1) Установите утилиту tcpdump. Просмотрите сетевой трафик вашего компьютера. Объясните, что показывает каждая строка в выводе.

1.2) Отфильтруйте трафик так, чтобы отображались только пакеты, отправляемые на определённый IP-адрес, например, 8.8.8.8:

1.3) Сохраните трафик в файл формата .pcap для последующего анализа:
Проверьте содержимое файла с помощью утилиты tcpdump -r.

2) Перенаправление ввода и вывода:

2.1) Перенаправьте результат команды ls -l /home в файл output.txt. Убедитесь, что в терминале ничего не выводится, а результат находится в файле.

2.2) Перенаправьте ошибку от команды ls, указав несуществующую директорию, в файл errors.txt.

2.3) Перенаправьте и стандартный вывод, и ошибки команды ping в один файл.

2.4) Используя tee, сохраните результат команды df -h одновременно в файл disk_usage.txt и отобразите его в терминале.

3. Анализ сети и соединений:

3.1) С помощью утилиты netstat или ss определите текущие сетевые соединения и выведите их в файл connections.txt.

3.2) Используйте команду ping для проверки доступности google.com. Сохраните результат в файл ping_google.txt. Затем добавьте к нему данные о времени выполнения команды.

3.3) С помощью утилиты traceroute определите маршрут до сервера 8.8.8.8 и сохраните результат в файл route_to_google.txt.

4) Практика работы с пайпами:

4.1) Выведите список файлов в каталоге /usr/bin, содержащих в имени "ssh", и сохраните его в файл ssh_files.txt.

4.2) Найдите строки, содержащие слово "error" в файле журнала /var/log/syslog, и перенаправьте результат в файл errors_found.txt

4.3) С подсчётом количества строк отфильтруйте строки, содержащие слово "warning" в том же файле:

Контрольные вопросы

1. Как с помощью tcpdump захватить только те пакеты, которые отправляются с вашего компьютера на определённый порт удалённого узла (например, порт 443)? Приведите полный пример команды и объясните её части.

2. Вы выполняете команду, которая одновременно выводит информацию и генерирует ошибки. Как записать стандартный вывод в один файл, а стандартные ошибки в другой, при этом сохранив возможность просмотра результата в терминале?

3. После смены сетевых настроек в системе вы не можете подключиться к удалённому серверу. Какие команды вы выполните для диагностики проблемы? Объясните порядок их использования и возможные результаты.

4. Как можно использовать tcpdump для захвата трафика, относящегося только к DNS-запросам, выполненным с вашего компьютера? Приведите пример команды и объясните, какие фильтры применяются для этого

5. Как организовать цепочку команд для выполнения следующего сценария:

1)Найти все строки, содержащие слово "error" в файле журнала /var/log/syslog.

2)Отфильтровать только те строки, которые содержат дату сегодняшнего дня.

3)Сохранить результат в файл и одновременно вывести количество таких строк в терминал.