**Министерство науки и высшего образования Российской Федерации** ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
 **НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Компьютерные сети»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

«Основы администрирования маршрутизируемых

компьютерных сетей»

**Выполнил:**

Арендаренко М. М., студент группы N3347

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Есипов Д.А

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Санкт-Петербург

2024 г.

# Содержание

Оглавление

[Содержание 2](#_Toc179507865)

[Введение 3](#_Toc179507866)

[1. Настройка сетевых интерфейсов 4](#_Toc179507867)

[2. Проверка соединения 7](#_Toc179507868)

[3. Настройка правил iptables 9](#_Toc179507869)

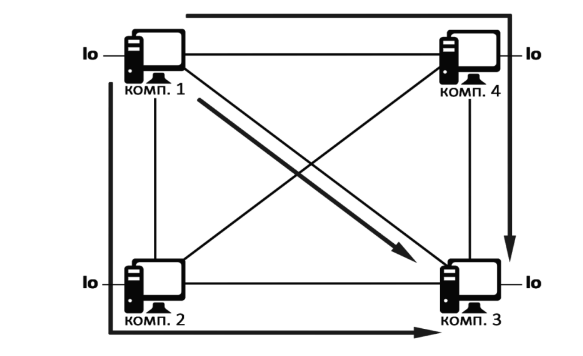
[Описание команд: 9](#_Toc179507870)

[**Заключение:** 13](#_Toc179507871)

# Введение

Цель работы – изучение основных методов настройки маршрутизируемых компьютерных сетей на примере сети, состоящей из компьютеров под управлением операционной системы Linux. В ходе выполнения задания изучаются различные аспекты сетевого уровня модели OSI, включая базовую настройку маршрутизации, управление таблицами маршрутов, а также правила трансляции сетевых адресов (NAT). Используется утилита tcpdump для наблюдения за сетевым трафиком и анализа структуры пакетов, проходящих через интерфейсы маршрутизируемых узлов.

# Настройка сетевых интерфейсов

Для выполнения работы был выбран вариант 2: 6 + (17 mod 5) = 8  


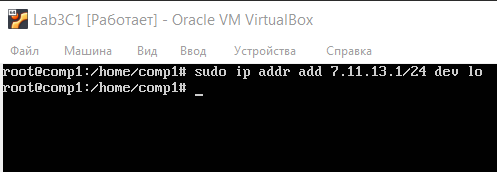


Рис. 1 Назначение IP адреса для компьютера 1

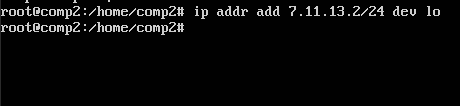
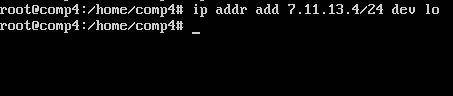


Рис. 2 Назначение IP адреса для компьютера 2



Рис. 3 Назначение IP адреса для компьютера 3



Настройка компьютера 1

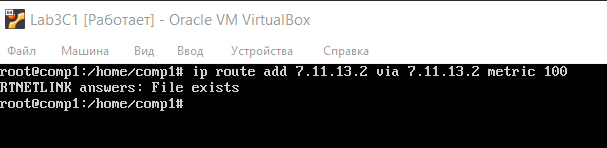


Рис. 4 Для маршрута через шлюз 7.11.13.2 (Компьютер 2)

Настройка маршрута для компьютера 2 через шлюз с IP-адресом 7.11.13.2. В команде используется вес (metric) 10 для приоритета данного маршрута.

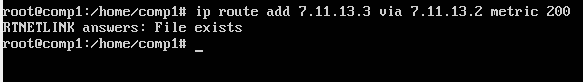


Рис. 5 Для маршрута через шлюз 7.11.13.5 (Компьютер 3)

Настройка маршрута для компьютера 3 через шлюз с IP-адресом 7.11.13.5. В команде используется вес (metric) 20, что устанавливает приоритет для данного маршрута ниже, чем у компьютера 2

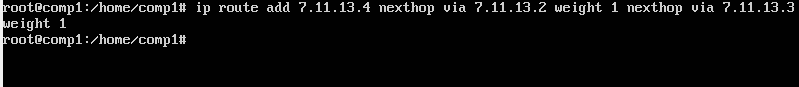


Рис. 6 Для настройки маршрута через компьютеры 2 и 3 (Компьютер 4)

Настройка маршрута для компьютера 4 с несколькими шлюзами. Маршрут проходит через компьютеры 2 и 3 с балансировкой нагрузки, где вес (weight) для каждого следующего хопа равен 1

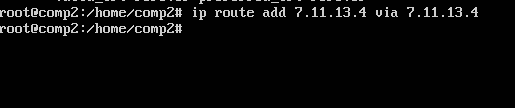
Настройка компьютера 2  


Рис. 7 Настройка маршрута для компьютера 4 через компьютер 2

Маршрут для компьютера 4 настраивается на компьютере 2. Команда создаёт маршрут, который направляет трафик к IP-адресу 7.11.13.4 через этот же адрес. Этот маршрут позволяет организовать связь между компьютерами в сети

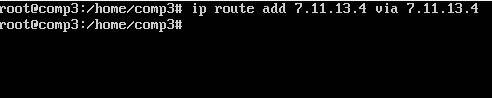


Рис. 8 Настройка маршрута для компьютера 4 через компьютер 3

Маршрут для компьютера 4 настраивается на компьютере 3. Команда создаёт маршрут, который направляет трафик к IP-адресу 7.11.13.4 через этот же адрес. Этот маршрут позволяет организовать связь между компьютерами в сети

# Проверка соединения

Перезапускам интерфейс на всех компьютерах для корректной работы и проверяем доступность компьютеров посредством команды **ping**

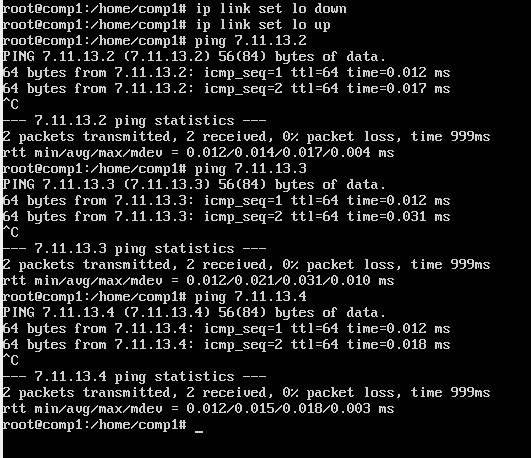


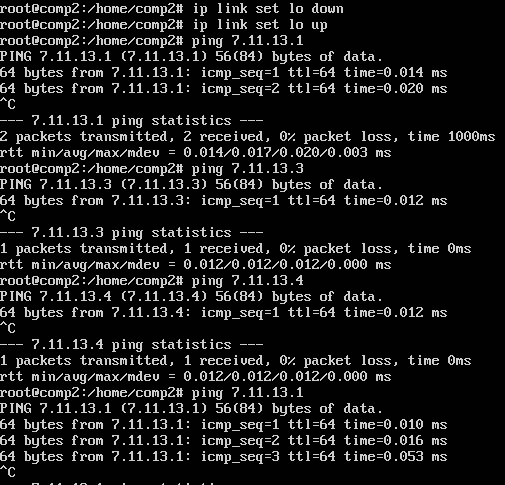
Рис. 9 Проверка соединения через компьютер 1  
  


Рис. 10 Проверка соединения через компьютер 2

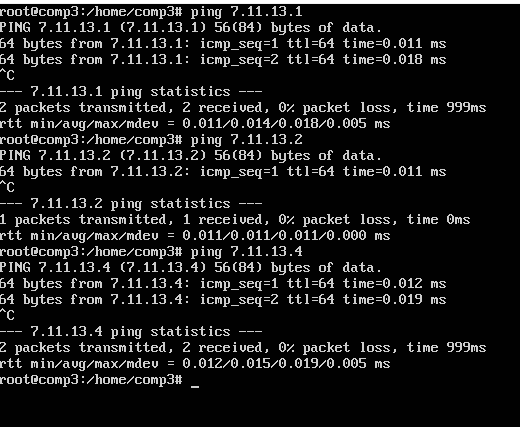


Рис. 11 Проверка соединения через компьютер 3

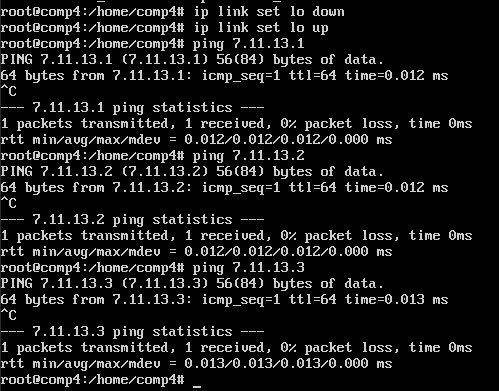


Рис. 12 Проверка соединения через компьютер 4

# Настройка правил iptables

Рис. 13 Выставление правил iptables

Описание команд:

1. **Запрет передачи пакетов на TCP-порт 12345 (используемый nc)**

|  |
| --- |
| sudo iptables -A OUTPUT -p tcp --dport 12345 -j DROP |

Эта команда добавляет правило в таблицу **OUTPUT**, которое блокирует исходящие пакеты, передаваемые по протоколу **TCP** на порт **12345**

* **-A OUTPUT:** Добавляет новое правило в цепочку OUTPUT, которая отвечает за обработку исходящего трафика.
* **-p tcp:** Указывает, что это правило применяется к трафику по протоколу TCP.
* **--dport 12345:** Это параметр задаёт целевой порт назначения, в данном случае — порт 12345.
* **-j DROP:** Указывает действие для пакетов, которые соответствуют правилу — в данном случае, пакеты будут отброшены (блокированы).

**2.** **Запрет приёма пакетов с UDP-порта 12345 (используемого nc)**

|  |
| --- |
| sudo iptables -A INPUT -p udp --sport 12345 -j DROP |

Эта команда добавляет правило в таблицу **INPUT**, которое блокирует входящие пакеты, передаваемые по протоколу **UDP,** исходящие с порта **12345.**

* **-A INPUT:** Добавляет новое правило в цепочку **INPUT,** которая отвечает за обработку входящего трафика.
* **-p udp:** Указывает, что это правило применяется к трафику по протоколу **UDP.**
* **--sport 12345:** Это параметр задаёт исходный порт отправителя, в данном случае — порт **12345.**
* **-j DROP:** Указывает действие для пакетов, которые соответствуют правилу — в данном случае, пакеты будут отброшены (блокированы).

**3. Запрет передачи пакетов с IP-адреса компьютера А (7.11.13.1)**

|  |
| --- |
| sudo iptables -A OUTPUT -s 7.11.13.1 -j DROP |

Эта команда добавляет правило в таблицу **OUTPUT**, которое блокирует исходящие пакеты, если они отправляются с IP-адреса **7.11.13.1**

* **-A OUTPUT:** Добавляет новое правило в цепочку **OUTPUT**, которая отвечает за исходящий трафик.
* **-s 7.11.13.1**: Указывает IP-адрес источника, откуда отправляются пакеты — в данном случае, это адрес **7.11.13.1.**
* **-j DROP:** Указывает действие — отброс пакетов (блокировка).

**4. Запрет приёма пакетов, отправленных на IP-адрес компьютера Б (7.11.13.2)**

|  |
| --- |
| sudo iptables -A INPUT -d 7.11.13.2 -j DROP |

Эта команда добавляет правило в таблицу **INPUT**, которое блокирует входящие пакеты, отправленные на IP-адрес **7.11.13.2.**

**-A INPUT:** Добавляет новое правило в цепочку **INPUT**, которая обрабатывает входящий трафик.

**-d 7.11.13.2:** Указывает целевой IP-адрес пакетов — в данном случае, **7.11.13.2.**

**-j DROP:** Указывает действие для таких пакетов — отброс (блокировка)

**5. Запрет приёма и передачи ICMP-пакетов с длиной более 1000 байт и TTL меньше 10  
 5.1** **Команда для входящих ICMP-пакетов:**

|  |
| --- |
| sudo iptables -A INPUT -p icmp --icmp-type echo-request -m ttl --ttl-lt 10 -j DROP |

Эта команда блокирует все входящие **ICMP-пакеты** типа **echo-request** (используемые командой **ping**), если их **TTL** меньше 10. Это предотвращает приём "старых" пакетов, которые прошли через большое количество маршрутизаторов.

* **-A INPUT**:
* Эта часть команды указывает, что новое правило добавляется в цепочку **INPUT**.
* Цепочка **INPUT** обрабатывает **входящие** пакеты, то есть те, которые поступают на текущий компьютер из сети.
* **-p icmp**:
  + Указывает, что правило применяется только к пакетам протокола **ICMP** (Internet Control Message Protocol). Этот протокол часто используется для отправки сообщений о состоянии сети, в частности для команд **ping**.
* **--icmp-type echo-request**:
  + Этот параметр уточняет, что правило касается только пакетов ICMP типа **echo-request**. Это специальные пакеты, которые используются в команде **ping** для проверки доступности хоста.
* **-m ttl**:
  + **-m** указывает на использование модуля **ttl**, который используется для фильтрации пакетов по их **TTL** (Time to Live).
* **--ttl-lt 10**:
  + Эта часть команды указывает, что правило будет применяться только к тем пакетам, у которых TTL **меньше 10**. TTL — это поле в IP-заголовке, которое определяет максимальное количество переходов (hops), через которые пакет может пройти. Когда TTL достигает 0, пакет отбрасывается. Чем меньше TTL, тем ближе пакет к завершению своего маршрута.
* **-j DROP**:
  + **-j** (jump) указывает, какое действие следует предпринять для пакетов, которые соответствуют критериям. В данном случае пакеты будут **отброшены** (DROP), то есть не обработаны и удалены.

**5.2** **Команда для исходящих ICMP-пакетов:**

|  |
| --- |
| sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -m ttl --ttl-lt 10 -j DROP |

Эта команда очень похожа на предыдущую, но вместо обработки входящих пакетов она действует на **исходящие** пакеты.

* **-A OUTPUT**:
  + Это правило добавляется в цепочку **OUTPUT**, которая отвечает за обработку **исходящего** трафика. Это пакеты, которые отправляет ваш компьютер в сеть.
* **-p icmp --icmp-type echo-request**:
  + Правило применяется к исходящим ICMP-пакетам типа echo-request (используемые командой ping).
* **-m ttl --ttl-lt 10**:
  + Правило будет действовать только для пакетов, у которых TTL **меньше 10**.
* **-j DROP**:
  + Исходящие пакеты, которые соответствуют этому правилу, будут **отброшены**.

Эта команда блокирует все исходящие ICMP-пакеты типа echo-request (используемые командой ping), если их TTL меньше 10. Это предотвращает отправку пакетов с очень низким TTL, которые почти наверняка будут отбрасываться по пути, если дойдут до конечного узла.

**Проверка правил iptables**

|  |
| --- |
| sudo iptables -L -v |

* -L: Список всех цепочек и их правил.
* -v: Включает детализированный вывод, который показывает, сколько пакетов обработано каждым правилом.

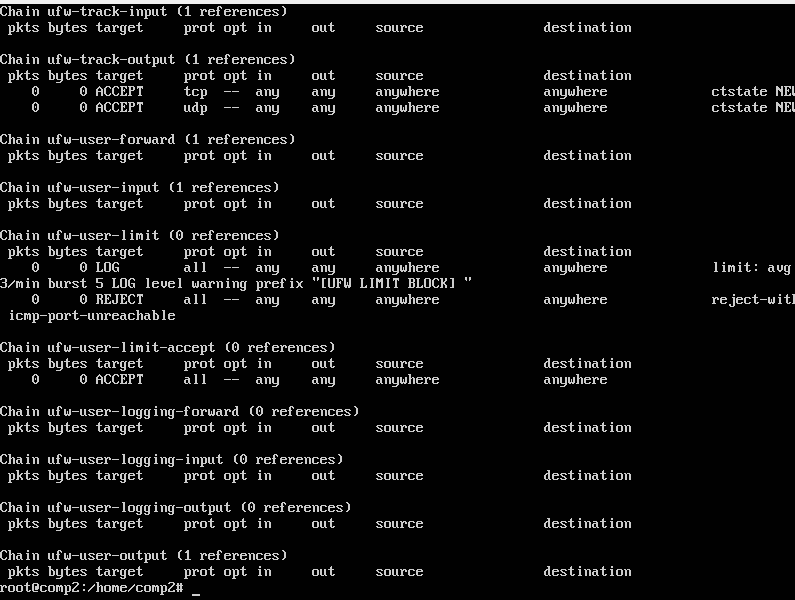


Рис. 14 Проверка наличия правил

Команда показала, что правила были успешно применены и сохранены

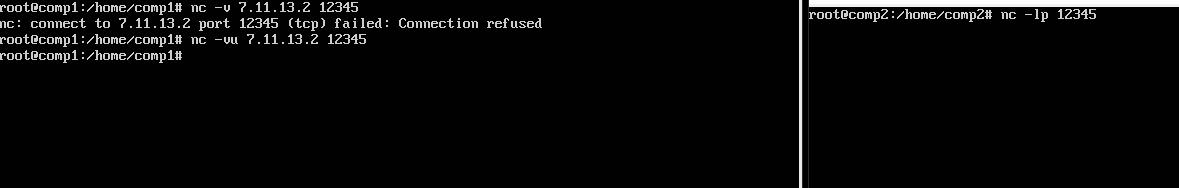
Проверяем их на примере команды nc:  
  


Рис. 15 Тестирование блокировок

Как мы можем видеть, мы не можем подключиться к другому компьютеру в сети, соответственно правила успешно работают

# **Заключение:**

В результате выполнения лабораторной работы были изучены и на практике реализованы основные методы настройки маршрутизируемых компьютерных сетей. В ходе работы особое внимание было уделено следующим аспектам:

1. **Базовая конфигурация маршрутизации**:
   * Настройка таблиц маршрутов с использованием утилиты ip позволила настроить маршрутизацию между несколькими хостами в сети. Были рассмотрены методы добавления статических маршрутов, а также их проверка с помощью утилит ping и traceroute. Это дало возможность лучше понять, как пакеты передаются между узлами через заданные маршруты и как корректно настраивать маршрутные таблицы.
2. **Управление сетевыми адресами**:
   * Важной частью работы стало назначение IP-адресов сетевым интерфейсам. Были проанализированы различные маски подсетей и их влияние на структуру сети. Работа с командами, такими как ip addr, позволила изучить настройку адресов для взаимодействия компьютеров в рамках одной подсети, а также между различными сетями.
3. **Работа с iptables и фильтрация пакетов**:
   * Были рассмотрены методы создания правил фильтрации трафика с использованием утилиты iptables. Настройка блокировок по портам и IP-адресам позволила ограничить доступ к определённым ресурсам сети. Это включало как входящие, так и исходящие соединения, что дало возможность научиться основам настройки сетевых экранов (firewalls) и обеспечению сетевой безопасности.
4. **Наблюдение за трафиком с помощью tcpdump**:
   * Использование утилиты tcpdump позволило на практике проанализировать сетевой трафик, проходящий через маршрутизируемые узлы. В частности, были рассмотрены пакеты ICMP, TCP и UDP, их заголовки и данные, что помогло глубже понять внутреннюю структуру сетевых пакетов. tcpdump был использован для отладки маршрутизации, поиска проблем с пакетами и наблюдения за изменениями в трафике после применения правил фильтрации.
5. **Технологии NAT и маршрутизация в IPv4**:
   * В ходе работы были также изучены основы технологии NAT (Network Address Translation), которая широко применяется для трансляции адресов в локальных сетях. Были продемонстрированы основные принципы работы NAT и его роль в маршрутизации пакетов между локальной сетью и внешними сетями. Основы работы с протоколом IPv4, включая назначение маршрутов, фильтрацию пакетов и управление адресами, являются ключевыми для администрирования современных компьютерных сетей.
6. **Тестирование настроек**:
   * После настройки маршрутизации и фильтрации трафика были проведены тесты с использованием утилит ping, traceroute, nc (netcat), которые подтвердили корректность работы настроенных маршрутов и правил фильтрации. Это позволило убедиться в правильности выполненных конфигураций и наглядно увидеть результаты их применения в сети.

В целом, выполненная работа обеспечила практическое знакомство с основными инструментами и методами настройки маршрутизируемых сетей на базе операционной системы Linux. Изученные навыки являются важными для сетевых администраторов, так как они позволяют гибко управлять маршрутизацией, сетевой безопасностью и анализировать трафик в реальных сетевых средах.