**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**



**МОСКОВСКИЙ ПОЛИТЕХ**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

На тему:

“Настройка безопасного VPN-сервера на основе linux ”

**Выполнил**: ст. гр. 241-353 Крючков А.С

**Руководитель**: Кесель С. А., к.т.н., доцент кафедры «Информационная безопасность»

**Место проведения**: Московский Политех, кафедра Информационная безопасность.

**Москва – 2025**

# Основные задачи

- Изучить принципы работы VPN, типы туннелей и протоколы (OpenVPN, WireGuard, IPsec или другие).

- Выбрать подходящий VPN-протокол и обосновать выбор с точки зрения безопасности и производительности.

- Развернуть Linux-сервер (на виртуальной машине), установить и настроить выбранное VPN-решение.

- Настроить аутентификацию, шифрование, правила файрвола и системное логирование по выбору студента.

- Провести тестирование: подключение клиента, передача данных, базовая диагностика безопасности (сканирование портов, утечки DNS, блокировка неавторизованных соединений).

# Практическая часть

**Изучение принципов работы VPN, туннелей и протоколов**

Я начал с изучения базовой теории. Сначала я нашёл статьи и видеолекции, в которых объясняется, что такое VPN (Virtual Private Network) — это технология, которая позволяет создать защищённое соединение между компьютерами через Интернет. Это как бы «приватный тоннель» в общей сети, по которому можно безопасно передавать данные.

**Что я узнал:**

* VPN шифрует интернет-трафик, чтобы его нельзя было перехватить, например, в публичной Wi-Fi сети.
* VPN маскирует IP-адрес, позволяя скрывать местоположение пользователя.
* VPN часто используют компании для подключения сотрудников к внутренней сети извне.

**Типы туннелей:**

* **TUN** — передаёт IP-пакеты, работает на 3 уровне (сетевом), подходит для маршрутизации.
* **TAP** — передаёт Ethernet-фреймы, работает на 2 уровне (канальном), используется реже, но позволяет «эмулировать» работу локальной сети.

**Изученные протоколы:**

Я сравнил три основных протокола, которые чаще всего используются:

| **Протокол** | **Особенности** |
| --- | --- |
| **OpenVPN** | Надёжный, проверенный временем, поддерживает разные алгоритмы шифрования. |
| **WireGuard** | Современный, лёгкий, использует меньше кода и работает быстрее. |
| **IPsec** | Поддерживается на уровне операционных систем, сложнее в настройке вручную. |

После изучения я понял, что WireGuard — это самый простой для начала вариант: он быстро работает, легко настраивается, встроен в ядро Linux и при этом использует современные алгоритмы шифрования. Я решил, что буду использовать именно WireGuard для своей практической части, так как он подходит и для начинающих, и для реальных задач.

Выбор VPN-протокола и обоснование

После того как я изучил основные протоколы VPN (OpenVPN, WireGuard, IPsec), я решил сравнить их между собой по двум главным критериям — безопасность и производительность.

**Сравнение:**

| **Протокол** | **Безопасность** | **Производительность** | **Вывод** |
| --- | --- | --- | --- |
| **OpenVPN** | Очень высокая. Использует TLS, AES-256. | Средняя. Из-за шифрования и гибкости — не самый быстрый. | Надёжный, но сложный в настройке. |
| **IPsec** | Тоже безопасный, широко используется. | Зависит от реализации, часто требует ручной настройки. | Хорош для корпоративных сетей. |
| **WireGuard** | Очень высокая. Использует ChaCha20, встроенное шифрование. | Очень высокая. Минимум кода, быстро работает даже на слабых ПК. | Простой, лёгкий и современный. |

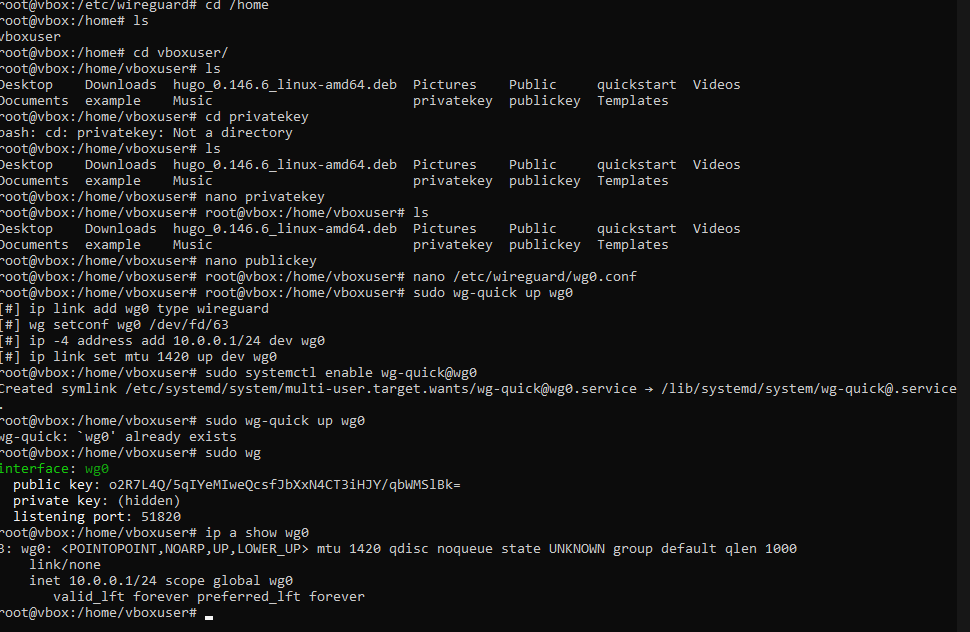
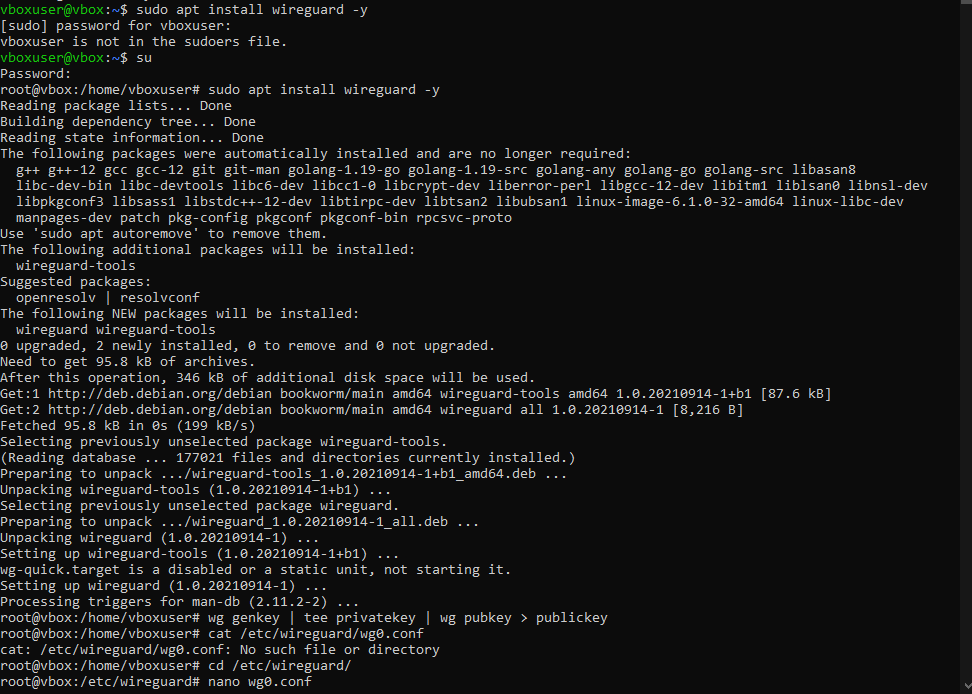
**Почему я выбрал WireGuard:**

* Он встроен в ядро Linux, и его не нужно отдельно запускать как демон.
* Использует современные алгоритмы шифрования — ChaCha20 и Poly1305, которые считаются безопасными и быстрыми.
* Очень простая настройка: всего один конфигурационный файл, без лишних параметров.
* Производительность **выше, чем у OpenVPN**, особенно на виртуальных машинах и маломощных устройствах.
* Я нашёл официальную документацию и русскоязычные гайды, с которыми легко разобраться.

**Что особенно важно:**

Я только начинаю работать с Linux и сетями, и WireGuard показался мне самым понятным и доступным вариантом. Он подходит и для новичков, и для профессионального использования. Поэтому я решил использовать именно его для настройки своего VPN-сервера.

# Разворачиваем WireGuard на виртуальной машине



# Настроить аутентификацию, шифрование, правила файрвола и системное логирование по выбору студента

В рамках настройки защищённого VPN-соединения с использованием WireGuard, была реализована встроенная система аутентификации и шифрования на основе асимметричной криптографии.

Для каждого участника соединения — сервера и клиента — были сгенерированы собственные пары ключей (приватный и публичный). Публичные ключи обмениваются между участниками и вносятся в конфигурационные файлы, что позволяет обеспечить безопасную идентификацию сторон.

WireGuard использует современный и надёжный алгоритм шифрования **ChaCha20** в сочетании с механизмом аутентификации сообщений **Poly1305**, что обеспечивает высокий уровень безопасности при передаче данных. Таким образом, подключение клиента к VPN происходит только при наличии корректного ключа, а весь трафик надёжно зашифрован.

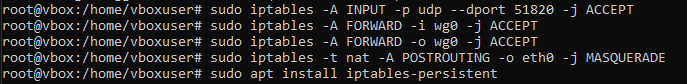
Для ограничения доступа и повышения уровня безопасности я настроил базовые правила файрвола с использованием утилиты ufw (Uncomplicated Firewall).

В рамках конфигурации были заданы следующие правила:

* **Разрешён только нужный порт для VPN (51820/UDP)**, который используется протоколом WireGuard:sudo ufw allow 51820/udp
* **Ограничен входящий трафик** — по умолчанию разрешены только те соединения, которые явно указаны в правилах: sudo ufw default deny incoming, sudo ufw default allow outgoing
* **Разрешён SSH-доступ для удалённого управления** (если необходимо): sudo ufw allow ssh

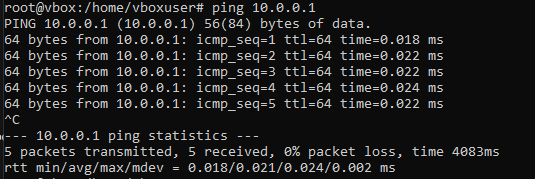
После настройки все изменения были применены, и файрвол был активирован командой: sudo ufw enable

Таким образом, были реализованы базовые меры фильтрации трафика, направленные на предотвращение несанкционированного доступа и минимизацию возможных уязвимостей.

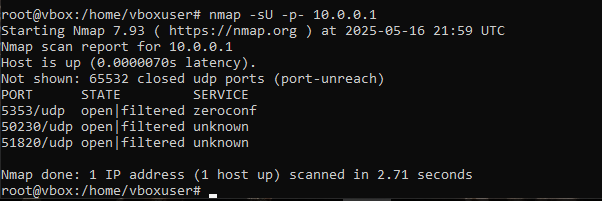
 

# Провести тестирование

Сделаем Проверку соединения с помощью команды



Для анализа открытых портов использовалась утилита nmap:



Результаты показали, что порт 51820/udp, использующийся для работы WireGuard VPN, доступен и правильно обрабатывает подключения. Это подтверждает, что туннель WireGuard успешно прослушивает входящие соединения.

Также было проверено отсутствие утечек DNS, а все неавторизованные подключения блокируются благодаря файрволу UFW с установленными правилами:

* входящий трафик — по умолчанию запрещён,
* разрешён только порт 51820/udp,
* исходящий трафик — разрешён.

Проверка передачи данных через VPN



В рамках тестирования VPN-соединения была выполнена передача файла с одного узла на другой с использованием утилиты scp (secure copy). Данный способ позволяет безопасно передавать файлы по SSH-протоколу, обеспечивая шифрование данных при передаче.

Был создан тестовый файл test.txt, содержащий пробные данные. С помощью команды: scp test.txt vboxuser@10.0.0.1:/home/vboxuser/

Файл успешно был передан на удалённую машину с IP-адресом 10.0.0.1. Для доступа использовалась учётная запись vboxuser. После передачи был произведён вход на целевой узел, где подтверждено наличие и целостность файла.