

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Российский экономический университет имени Г.В. Плеханова»  
**Московский приборостроительный техникум**

---

Выпускная квалификационная работа  
(Дипломная работа)

На тему: «Развертывание технологии Open VPN в корпоративной сети.»

Кочаряна Эрика Робертовича  
Студента 4 курса группы КС-3-17

по специальности: 09.02.02 Компьютерные сети  
для присвоения квалификации: техник по компьютерным сетям  
Форма обучения: очная

Руководитель: \_\_\_\_\_/Чурилова Т.Ю./  
«\_\_\_\_» \_\_\_\_\_ 2021 г.

Студент: \_\_\_\_\_/Кочарян Э. Р./  
«\_\_\_\_» \_\_\_\_\_ 2021 г.

Допущен к защите  
Распоряжением от «18» мая 2021 г. №18.01-11-260

Москва, 2021 г.

## Содержание.

Введение.....	2
Основная часть. ....	5
Раздел 1 «Теоретическая часть».....	5
Глава 1 Ознакомление с технологией VPN.....	5
Задача 1.1 Терминология .....	5
Задача 1.2 Принцип работы VPN .....	7
Глава 2. Ознакомление с протоколом Open VPN.....	11
Задача 2.1 Протокол Open VPN.....	11
Задача 2.2 Актуальность Open VPN в сравнение с протоколом WireGuard .....	16
Раздел 2 «Практическая часть» .....	19
Глава 1. Постановка практической задачи, подготовка топологии к внедрению протокола OpenVPN. ....	19
Задача 1.1 Построение топологии.....	19
Задача 1.2 Настройка сетевых устройств .....	21
Глава 2. Развёртывание протокола Open VPN.....	23
Задача 2.1 Создание центра сертификации. Развёртывание Open VPN сервера для site-to-site соединения. ....	23
Задача 2.2 Конфигурирование и подключение Open VPN клиентов.....	29
Задача 2.3 Проверка работоспособности Open VPN.....	36
Заключение .....	38
Список используемых источников и литературы.....	40

## **ВВЕДЕНИЕ**

В последние пару лет VPN всё чаще оказывается на слуху. Это происходит по разным причинам, например, из-за суровых законов некоторых стран в отношении интернет-ресурсов, а также из-за пандемии коронавируса из-за, которой сотрудники различных компании все чаще уходят на удаленную работу.

VPN протоколы активно используется для решения самых разных задач от обхода блокировок сайтов на территории определенных стран, до объединения филиалов компании в единую сеть. Также часто VPN применяется в компаниях для безопасного удаленного подключения сотрудника к локальной сети предприятия. Это основные сценарии использования VPN.

В данной выпускной квалификационной работе будет рассматриваться тема “Развертывание технологии Open VPN в корпоративной сети”.

В теоретической части квалификационной работы рассматриваются такие темы как:

- Основные термины связанные с VPN (инкапсуляция, туннель, шифр и т.д.). Принцип работы туннелирования и VPN, в частности.
- Протокол Open VPN. Сертификаты и центр сертификации. Варианты развертывание технологии Open VPN в корпоративной сети.
- Актуальность протокола Open VPN в сравнение с новым протоколом WireGuard.

В практической части работы будет рассмотрена настройка и развертывание протокола Open VPN в корпоративной сети в рамках проекта в программе GNS3.

Понимание принципов работы VPN очень важно для современного специалиста в области компьютерных сетей в современном мире.

Протокол Open VPN считается одним из самых популярных протоколов на данный момент. По многим причинам, которые будут рассмотрены в данной выпускной квалификационной работе.

Тема данной работы является актуальной и рекомендуется к прочтению специалистам в области системного администрирования.

## **Основная часть.**

### **Раздел 1 Теоретическая часть.**

#### **Глава 1 Ознакомление с технологией VPN.**

На данный момент существует множество различных VPN протоколов, различающихся по самым разным параметрам таким как: безопасность, поддерживаемость различными ОС, скорость работы и тд. Для решения разных задач используются разные протоколы. Существуют, как и устаревшие протоколы “канувшие в небытие” так и новые активно разрабатываемые и набирающие популярность. Но все VPN протоколы работают по определенному принципу.

Чтобы понять принцип работы VPN и понимать то, о чём будет говориться в данной выпускной квалификационной работе необходимо знать основную связанную с VPN терминологию.

В рамках данной главы будут даны определения терминам связанным с VPN, а также будет рассмотрен принцип работы VPN.

##### **1.1 Терминология.**

**VPN (Virtual Private Network)** – Виртуальная частная сеть. Эта технология позволяющая создать поверх публичной сети, такой как Интернет, другую защищенную сеть с ограниченным числом участников в сети. VPN является защищенной сетью благодаря использованию различных средств, криптографии.[4]

**Туннелирование** – Это процесс создание логического соединения между двумя конечными точками посредством инкапсуляции различных протоколов. Туннелирование это метод построения сетей, при котором один сетевой протокол инкапсулируется в другой.[14]

**Инкапсуляция** – Это процесс формирования пакета перед отправкой по сети.[8] Инкапсуляция происходит сверху вниз по модели (OSI) начиная с прикладного уровня и заканчивая физическим уровнем. На каждом уровне к пакету добавляется новый блок информации, предназначенный для конкретного уровня. Также есть и обратный процесс **деинкапсуляция**.

**Туннель (Point-to-point)** - Зашифрованное соединение между клиентом и сервером. Туннель создаётся в незащищённой сети, в качестве которой чаще всего выступает Интернет. Соединение «точка-точка» подразумевает, что оно всегда устанавливается между двумя компьютерами, которые называются «узлами» или «peers». Каждый «peer» отвечает за шифрование данных до того, как они попадут в туннель, и расшифровка этих данных произойдёт после того, как они покинут туннель.[16]

**Шифр (Cipher)** – Математический алгоритм, используемый для шифрования данных.[2]

**Порт** – Это некоторое целое число, используемое в заголовках протоколов транспортного уровня модели OSI.[3] Порты необходимы для возможности взаимодействия разных протоколов (программ) с одним IP адресом.

**Аутентификация** — это процесс проверки подлинности чего-либо [1]. К примеру, проверка введенного пользователем пароля и пароля, находящегося в базе данных сервера.

## 1.1 Принцип работы VPN.

Чтобы понять принцип работы VPN, необходимо для начала понять, как инкапсулируется пакет с целью создания туннеля. Рассмотрим структуру пакета (Рис.1). Вначале идёт заголовок Ethernet в нем указывается мак адрес отправителя и мак адрес получателя данных данный заголовок соответствует канальному уровню модели OSI, также в этом заголовке указывается, какой протокол используется на сетевом уровне обычно это протокол IP (IPv4 или IPv6).[7] Далее идёт IP заголовок, в нём указаны IP адреса (получателя, отправителя), а также информация о том какой протокол используется на транспортном уровне UDP или TCP. После идёт заголовок TCP или UDP в нём указаны порт отправки и порт получения пакета, а также различная служебная информация.



Рисунок 1 – Структура пакета.

Для доставки пакета от одного узла до другого узла достаточно этих трёх уровней. Для доставки пакета, как правило, промежуточным устройствам нет необходимости «рассматривать» пакет «глубже». С тем что лежит внутри заголовка TCP это «личное дело» отправителя пакета и получателя пакета.

В стандартных случаях внутри, например, заголовка TCP идёт информация, допустим о протоколе HTTP.[5] Но при использовании туннелирования внутри заголовка TCP записывается ещё один заголовок Ethernet, а внутри него IP заголовок, а внутри него UDP заголовок и тд. Для удобства назовем эту запись «внутренним» пакетом.

Таким образом, получается пакет внутри пакета. Что это даёт? А именно то, что при деинкапсуляции пакета, отправленного по глобальной сети, происходит доставка «внутреннего» пакета, у которого так называемый виртуальный IP адрес. [6] Благодаря этому виртуальному адресу ПК, участвующие в процессе обмена пакетами «думают» что находятся в одной сети, то есть рядом. Таким образом, появляется виртуальная сеть поверх глобальной сети Интернет между двумя узлами. Также на уровне «внутреннего» пакета путь до получателя выглядит как один шаг, хотя в процессе передачи этого пакета происходит множество шагов в сети Интернет.

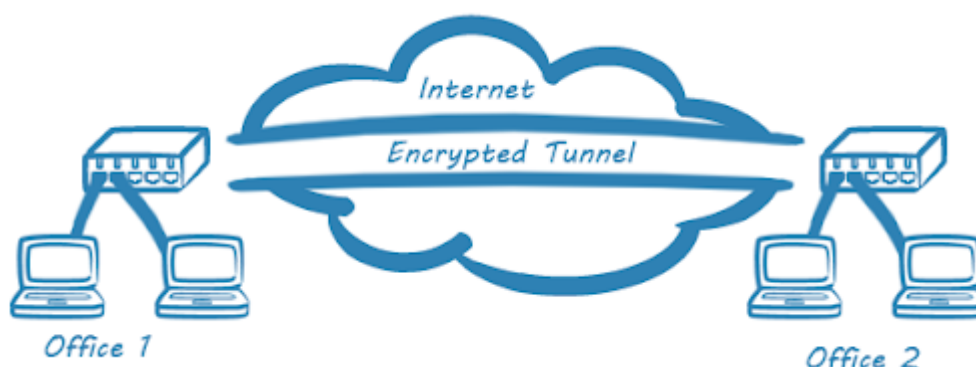
При добавлении шифрования и аутентификации к туннелированию получается VPN.[9]

**Протокол VPN** – основа любого VPN сервиса [11]. Протокол в данном случае является фундаментом, на котором выстроен сервис, ведь в нем содержатся протоколы передачи данных и стандарты шифрования, которые позволяют быстро и защищено обмениваться данными с VPN-серверами.

В зависимости от требований и применяемых протоколов, VPN может быть трех видов: сеть в сеть (site-to-site), удаленный доступ (Point-to-site), клиент/сервер (client/server). [12]

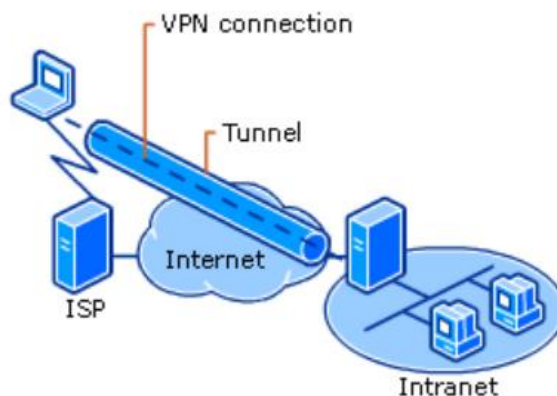


**Сеть в сеть (site-to-site)** – Объединение двух и более локальных сетей в единую виртуальную сеть.[12] Такой тип соединения часто используется для объединения сетей филиалов предприятия (Рис.2). Данный тип соединения очень удобен если филиалы компании достаточно сильно удалены друг от друга. Это соединения намного дешевле и надежнее проведения оптоволоконной линии связи.



*Рисунок 2 – Соединение site-to-site VPN.*

**Удаленный доступ (Point-to-site)** – Подключение типа точка сеть позволяет создать безопасное соединение отдельного компьютера (узла) с виртуальной сетью.[10] Такой типа соединения в основном используется для предоставления доступа удаленному сотруднику к локальным ресурсам сети предприятия, к примеру общей папке Windows.



*Рисунок 3 – Соединение типа Point-to-site.*

**Client-server VPN** - Этот вариант обеспечивает защиту передаваемых данных между двумя узлами (не сетями) корпоративной сети.[19] Особенность данного варианта в том, что VPN строится между узлами, находящимися, как правило, в одном сегменте сети, например, между рабочей станцией и сервером. Такая необходимость очень часто возникает в тех случаях, когда в одной физической сети необходимо создать несколько логических сетей. Например, когда надо разделить трафик между финансовым департаментом и отделом кадров, обращающихся к серверам, находящимся в одном физическом сегменте. Этот вариант похож на технологию VLAN, но вместо разделения трафика используется его шифрование.[17]

## **Глава 2 Ознакомление с протоколом Open VPN.**

### **2.1 Протокол Open VPN.**

**Open VPN** – протокол VPN с открытым исходным кодом выпущенный в 2002 году и активно развивающийся по сегодняшний день.[16] На данный момент один из самых популярных VPN протоколов. Благодаря открытому исходному коду протокол прошел большое количество проверок безопасности от компания занимающихся информационной безопасностью

Open VPN считается одним из самых защищенных протоколов, этому поспособствовало множество поддерживаемых алгоритмов шифрования, поддержка SSL, а также алгоритмов AES-256-GCM.[18]

Протокол OpenVPN является одним из самых быстрых протоколов многие VPN сервисы, использующие за основу Open VPN способны обеспечить скорость передачи зашифрованных данных до 2000 Мбит в секунду. Но в тоже время Open VPN нельзя назвать самым быстрым VPN протоколом быстрее его работает такой протокол как WireGuard.

Open VPN может работать на всех современных ОС, но только после установки специальной программы клиента Open VPN.[17]

#### **Обмен данными по сети.**

Open VPN способен передавать данные по сети на любом порту TCP или UDP с применением драйверов TUN/TAP.[15] Протокол UDP в сочетании с драйвером TUN позволяет подключиться клиенту находящемуся за NAT, то есть с серым IP адресом подключиться к серверу Open VPN.

Драйвер TAP эмулирует Ethernet устройство работающие на канальном уровне модели OSI, а драйвер TUN (сетевой туннель) на сетевом уровне модели OSI.

Возможность Open VPN выбирать любой порт, позволяет преодолевать ограничения маршрутизаторов и фаерволлов провайдера в случае их наличия.

### **Безопасность и шифрование.**

В Open VPN для обеспечения безопасности и шифрования используется библиотека OpenSSL в сочетании с протоколом транспортного уровня TLS который пришел на смену протоколу SSL.

В OpenSSL есть возможность использовать как симметричное, так и асимметричное шифрование.

Симметричное шифрование — это когда для шифрования и расшифрования используется один и тот же ключ.[1] В данном варианте ключ должен быть известен обоим сторонам в чём и зачастую заключается проблема при начальной передаче ключа одной из сторон (этап согласования ключей). Если злоумышленник получит этот ключ, то он будет иметь доступ ко всей информации передаваемой по каналу связи.

Ассимметричное шифрование — в данном случае используется два ключа закрытый и открытый.[1] Открытый ключ передается по открытым каналам связи и используется для шифрования. А для расшифровки используется закрытый ключ. Таким образом если кто-то перехватит открытый ключ, то он не сможет изменить передаваемую информацию или прочитать.

В основном при использовании OpenSSL для генерации ключей используют асимметричное шифрование. Генерируется два ключа публичный для шифрования данных и приватный для расшифровки данных.

Приватный ключ нельзя передавать на другие компьютеры кроме того на котором он создан. Можно передавать только публичный ключ.

Для безопасной передачи данных вначале нужно идентифицировать участников обмена данными иначе есть вероятность стать жертвой атаки называемой “человек по середине”. В ходе такой атаки злоумышленник подключается к каналу связи и прослушивает его. Он может спокойно перехватывать данные и даже изменять или удалять.

Для защиты от этой атаки используется проверка подлинности пользователя (аутентификация), для этого используется протокол TLS и инфраструктура открытых ключей (PKI - Public Key Infrastructure) в сочетании с асимметричной криптографией.[19]

Существует возможность расшифрования данных без наличия приватного ключа, например, методом перебора. Размер ключа хоть и влияет на сложность расшифровки, но в тоже время замедляет процесс обмена данными.

### **Сертификаты и центр сертификации СА**

Для избежание подделки открытого ключа используется процесс заверения ключа. Если ключ будет создан для публичного использования, то его должна заверить коммерческая или государственная организация с чистой репутацией. В результате процедуры заверения ключа создается сертификат открытого ключа. [14]

Созданный сертификат также должна заверить организация которой доверяют.

Есть много организаций, создающих сертификаты, к примеру для протокола HTTPS или для цифровой подписи сообщений электронной почты. Такие сертификаты стоят денег и служат ограниченный срок.

Для сети VPN используемой в конкретной компании нет необходимости покупать сертификаты. Можно создать свой собственный центр сертификации CA и создавать собственные сертификаты. Естественно доверие к таким сертификатам будет только на уровне той организации, где был создан этот сертификат. Основными плюсами использования собственного центра сертификации CA можно считать его абсолютную бесплатность и то, что такого CA будет вполне достаточно для создания виртуальной частной сети VPN на базе протокола Open VPN.

Созданные сертификаты будут играть роль открытых ключей с помощью, которых клиенты сети Open VPN будут шифровать данные. Соответственно для расшифровки данных будут использоваться приватные ключи.

Сертификаты создаются в соответствии со стандартом X.509. Этот стандарт определяет тип данных и процессы распределения открытых ключей с помощью сертификатов, снабженных электронными подписями.

Сертификат X.509 – это публичный ключ, в котором содержатся следующие данные: владелец сертификата, имя узла, период действия, алгоритм и значение подписи сертификата и т.д. [17] Он должен быть подписан закрытым ключом центра сертификации.

Когда клиент Open VPN подключается к серверу Open VPN по протоколу TLS то происходит отправка ему с сервера сертификата X.509. Для проверки подписи используется открытый ключ центра сертификации, находящийся на клиенте. Таким образом происходит проверка сервера, к которому подключается клиент это позволяет избежать атаки “человек по середине”. [8]

### **Список отзыва сертификатов**

В Open VPN предусмотрен список отзыва сертификатов (CRL), а также средства управления этим списком. Данный список необходим если нужно заблокировать доступ к VPN какому-либо отдельному сотруднику (клиенту), к примеру уволившемуся.

Список CRL генерируется в центре сертификации CA и потом копируется на сервер Open VPN. После изменения списка CRL его нужно заново копировать на сервер Open VPN.

### **Файл Диффи-Хелмана**

Файл Диффи-Хелмана в протоколе Open VPN используется для обеспечения защиты перехваченного трафика от расшифровки, при утере ключей. Под перехваченным трафиком имеется введенный записанный трафик до похищения ключей.

Данный файл создается на сервере Open VPN, а не на сервере центра сертификации.

## Статический ключ HMAC

HMAC (Hash-based Message Authentication Code) – обеспечивает проверку подлинности информации, передаваемой между сторонами. Этот ключ генерируется на сервере Open VPN с целью защиты от DoS атак.[16]

Пример файлов конфигурации клиента и сервера Open VPN рассматриваются в практической части.

### 2.2 Актуальность протокола Open VPN в сравнение с новым протоколом WireGuard.

**WireGuard** – Абсолютно новый VPN протокол с открытым исходным кодом визитной карточкой которого стал размер в 4000 строк. Активно набирает популярность у пользователей считается что в скором времени станет заменой OpenVPN. [15]

Для обеспечения безопасности WireGuard использует современную криптографию различных протоколов, берущих каждый на себя решение конкретной задачи:

- Curve25519 – для обмена ключами;
- Poly1305 – для аутентификации;
- ChaCha20 – для шифрования;
- SipHash – для ключей хеш-таблицы;
- BLAKE2 – для хеширования.

За все время существования этого протокола не было найдено серьезных уязвимостей. В случае же нахождения уязвимости разработчики могут очень быстро и оперативно исправить её, благодаря



небольшому объему кода. Также совсем не давно WireGuard был включен в состав ядра Linux 5.6 и прошел дополнительный аудит безопасности от независимых экспертов который не выявил не каких проблем с безопасностью.

По результату тестов производительности, которые можно посмотреть как на официальном сайте, так и у независимых тестировщиков можно прийти к выводу что по скорости работы WireGuard намного превосходит все ранее существующие VPN протоколы.[15]

При построении туннеля WireGuard использует UDP с выбором любого порта.

Хотя туннель WireGuard взломать можно сказать и невозможно, но если говорить о конфиденциальность, то WireGuard не как не защищает пользователей. Это происходит из-за того, что в настройках WireGuard в явном виде указываются IP адреса пользователей. И если сервером интересуются правоохранительные органы, то не кто не уйдет от внимания. Но трафик, конечно, останется неизвестным. В вопросе конфиденциальности WireGuard проигрывает OpenVPN, так как OpenVPN не требует от пользователей прописывать IP адреса в явном виде.

У WireGuard есть еще один минус в сравнении с OpenVPN это невозможность использовать протокол TCP что в свою очередь дает меньшего маневра в случае блокировки провайдером определенного трафика. К примеру, в случае с Open VPN всегда можно использовать 443 TCP порт, на котором работает HTTPS, провайдер просто не сможет его заблокировать поскольку тогда у пользователей не будут открываться сайты в Интернете.

Если говорить о минусах WireGuard, то также можно и отметить ограниченность в выборе протоколов шифрования одним протоколом. В свою очередь OpenVPN может использовать разные алгоритмы шифрования.

Подводя итог стоит сказать, что у WireGuard действительно есть все шансы заменить протокол OpenVPN, но все же тех возможностей, которые сейчас он предоставляет недостаточно чтобы полностью заменить OpenVPN, хотя для решения многих задач WireGuard подойдет и будет даже лучше других протоколов. [15]

Для организации корпоративного VPN все-таки лучше использовать OpenVPN введу его большей мобильность и защищенности.

## **Раздел 2. Практическая часть.**

### **Глава 1. Постановка практической задачи, подготовка топологии к внедрению протокола OpenVPN.**

#### **1.1 Построение топологии.**

В рамках практической части рассматривается проект сети в программе GNS3 с использованием VirtualBox для эмуляции серверных и пользовательских ОС. В данном проекте используются прошивки роутеров cisco IOS c7200. Виртуальные машины Linux на базе Ubuntu Server 20.04, виртуальные машины Windows Server на базе Windows server 2019 ОС. Также в топологии есть клиентский ПК с Windows 10.

Сценарий: есть некоторая компания, имеющая офис в Москве, которая открыла филиал в Санкт-Петербурге. Данной компании необходим Site-to-Site OpenVPN для объединения сетей офиса и филиала. Также планируется использовать VPN для подключения удаленных сотрудников необходимо сделать так чтобы удаленных сотрудник имел доступ к ресурсам обеих сетей компании.

Проектирование топологии (Рис.4). Сеть главного офиса имеет маршрутизатор cisco c7200 данный маршрутизатор стоит на границе сети и является шлюзом в Интернет. В сети есть коммутатор (в данном случае используется программный коммутатор GNS3 его функций достаточно для решения необходимой задачи) к нему подключены два сервера один DC1 – Windows Server и второй Ubuntu Server который является, в частности, OpenVPN сервером эти два сервера находятся в разных VLAN с целью показать более сложный вариант настройки протокола OpenVPN.[20] Сеть филиала построена аналогично сети главного офиса за исключением того, что второй Ubuntu Server выступает в роли клиента

OpenVPN. Третьей локальной сетью в топологии является домашняя сеть удаленного сотрудника она состоит из маршрутизатора и клиентского ПК на базе Windows 10.

В центре топологии установлен маршрутизатор (ISP) с прошивкой cisco IOS c7200 выполняющий роль провайдера. IP адреса используемые в сети указаны на топологии (Рис.4). На граничных маршрутизаторах настроен NAT для выхода устройств в интернет, а также заданы на интерфейсах IP адреса.

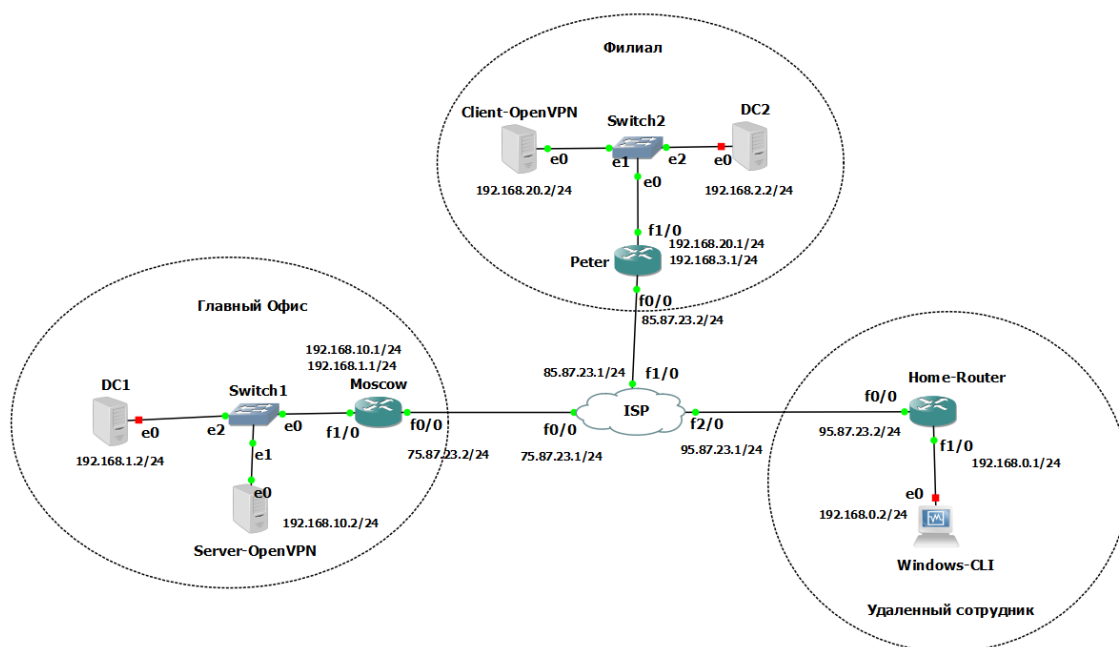


Рисунок 4 – Топология сети.

Оба Ubuntu 20.04 Server имеют hostname supernova. Для удобства была создана копия виртуальной машины.

## 1.2 Настройка сетевых устройств.

Первой проблемой при рассмотрении варианта развертывания OpenVPN в локальной сети за маршрутизатором станет невозможность клиентов достигать до сервера поскольку тот имеет серый IP адрес. Для того чтобы сервер имел белый IP адрес необходимо настроить проброс портов с помощью технологии NAT. Для этого используем выданные провайдером IP адреса и пробросим UDP порт 1194 к локальному адресу сервера (Рис.5) (именно этот порт будем использовать в OpenVPN).

```
ip nat inside source static udp 192.168.10.2 1194 75.87.23.10 1194 extendable
```

*Рисунок 5 – Проброс портов к клиенту и серверу OpenVPN.*

Перенос конфигурации с сервера на клиент — это вторая проблема, которая возникла в процессе выполнения данной работы. Поскольку Интернет-сервисов нету в данной сети, к примеру почты, а передать конфигурацию на флэшке мало того, что не возможно в рамках проекта GNS3 так еще и не логично так передавать информацию в реальной жизни так как филиал находится в другом городе. Для решения данной проблемы воспользуемся протоколом SFTP, но для этого для начала необходимо пробросить 22 TCP порт (ssh) к серверу и клиенту OpenVPN (Рис.6-Рис.7).

```
ip nat inside source static tcp 192.168.10.2 22 75.87.23.10 22 extendable
```

*Рисунок 6 – Проброс 22 порта до сервера.*

```
ip nat inside source static tcp 192.168.20.2 22 85.87.23.10 22 extendable
```

*Рисунок 7 – Проброс 22 порта до клиента.*

В данном варианте дизайна сети (Рис.4) трафик не будет идти через VPN сервер, а шлюзом в интернет служат в локальных сетях отдельные маршрутизаторы. Так как данные маршрутизаторы выступают в роли шлюза в интернет для все устройств локальной сети то необходимо на них прописать статические маршруты к серверу и клиенту OpenVPN соответственно (Рис.8-9) Для того, чтобы маршрутизатор знал куда отправить пришедший к нему пакет. В свою очередь машины с настроенным на них OpenVPN будут знать куда отправить пришедшие к ним пакеты.

```
ip route 0.0.0.0 0.0.0.0 75.87.23.1
ip route 10.10.10.0 255.255.255.0 192.168.10.2
ip route 192.168.2.0 255.255.255.0 192.168.10.2
ip route 192.168.20.0 255.255.255.0 192.168.10.2
```

*Рисунок 8 – Статические маршруты на маршрутизаторе главного офиса.*

На (Рис.8) указаны маршруты до сетей филиала, а также маршрут до сети OpenVPN в диагностических целях.

На (Рис.9) указаны маршруты до сетей главного офиса, а также до сети OpenVPN.

```
ip route 10.10.10.0 255.255.255.0 192.168.20.2
ip route 192.168.1.0 255.255.255.0 192.168.20.2
ip route 192.168.10.0 255.255.255.0 192.168.20.2
```

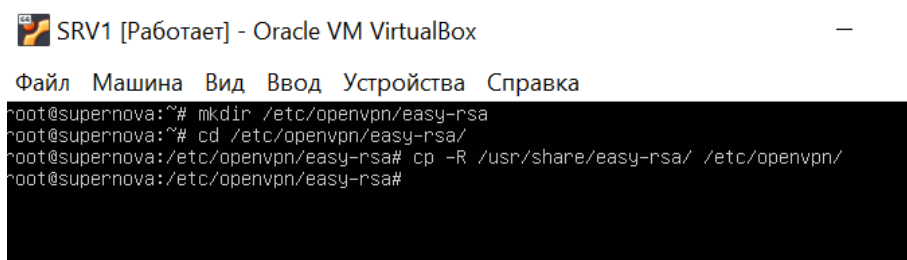
*Рисунок 9 – Статические маршруты на маршрутизаторе филиала.*

## Глава 2. Развертывание протокола OpenVPN.

### 2.1 Создание центра сертификации, генерация необходимых ключей. Запуск OpenVPN сервера.

Первое что необходимо сделать это установить необходимые пакеты, а именно “easy-rsa” и “openvpn” это можно сделать, используя команду “apt install”.

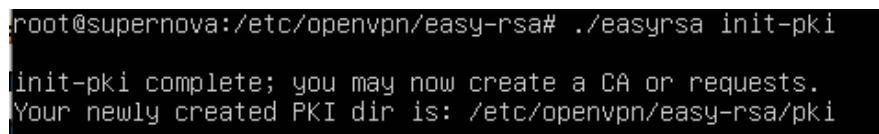
Создание центра сертификации. Для начала необходимо создать директорию, где будет находиться центр сертификации в данном случае центр сертификации находится на одном сервере с OpenVPN, поэтому для удобства расположим центр сертификации в директории, где будут находиться настройки OpenVPN (Рис.10).



```
SRV1 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
root@supernova:~# mkdir /etc/openvpn/easy-rsa
root@supernova:~# cd /etc/openvpn/easy-rsa/
root@supernova:/etc/openvpn/easy-rsa# cp -R /usr/share/easy-rsa/ /etc/openvpn/
root@supernova:/etc/openvpn/easy-rsa#
```

*Рисунок 10 Создание директории.*

Теперь произведем инициализацию центра сертификации, используя команду “./easyrsa init-pki” (Рис.11).



```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa init-pki
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/pki
```

*Рисунок 11 – Инициализация центра сертификации.*

После создадим корневую пару ключей командой: “./easyrsa build-sa”. При создании закрытого ключа центра сертификации предлагается ввести пароль. Стоит ввести пароль, так как если закрытый ключ попадет злоумышленнику, то это может привести к компрометации всех открытых ключей и сертификатов. Также указываем в поле “Common





Поскольку будет использоваться TLS авторизация то для предотвращения DoS атак по протоколу UDP необходимо сгенерировать ключ Hash-based Message Authentication Code (HMAC) (Рис.13). Данный ключ будет расположен также в директории “pki”.

```
root@supernova:/etc/openvpn/easy-rsa# openvpn --genkey --secret pki/ta.key
root@supernova:/etc/openvpn/easy-rsa#
```

*Рисунок 13 – Создание ключа HMAC.*

Для отзыва уже сгенерированных сертификатов необходимо создать сертификат отзыва (Рис.14). Сертификат отзыва необходим для возможности принудительного отключение от частной сети клиента, к примеру, уволенных сотрудников компании. Ключ “crl.pem” создается в директории “pki”.

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa gen-crl

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:

An updated CRL has been created.
CRL file: /etc/openvpn/easy-rsa/pki/crl.pem
```

*Рисунок 14 – Создание сертификата отзыва.*

Теперь можно перейти к созданию сертификатов используемых OpenVPN сервером. Командой “./easyrsa build-server-full supernova nopass” создаются все необходимы ключи для настройки OpenVPN (Рис.15).

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-server-full supernova nopass

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/supernova.key.gmyQF1DYz4'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'supernova'
Certificate is to be certified until Apr 20 10:12:07 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
root@supernova:/etc/openvpn/easy-rsa#
```

*Рисунок 15 – Создание сертификатов для сервера.*

На данном этапе все ключи и сертификаты для настройки сервера созданы и можно перейти, к конфигурированию сервера, но для начала скопируем ключи в директорию, где будет лежать конфигурация сервера для удобства прописывания путей до ключей и сертификатов (Рис.16).

```
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/ca.crt /etc/openvpn/ca.crt
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/dh.pem /etc/openvpn/dh.pem
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/crl.pem /etc/openvpn/crl.pem
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/ta.key /etc/openvpn/ta.key
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/issued/supernova.crt /etc/openvpn/supernova.crt
root@supernova:/etc/openvpn/easy-rsa# cp ./pki/private/supernova.key /etc/openvpn/supernova.key
root@supernova:/etc/openvpn/easy-rsa# _
```

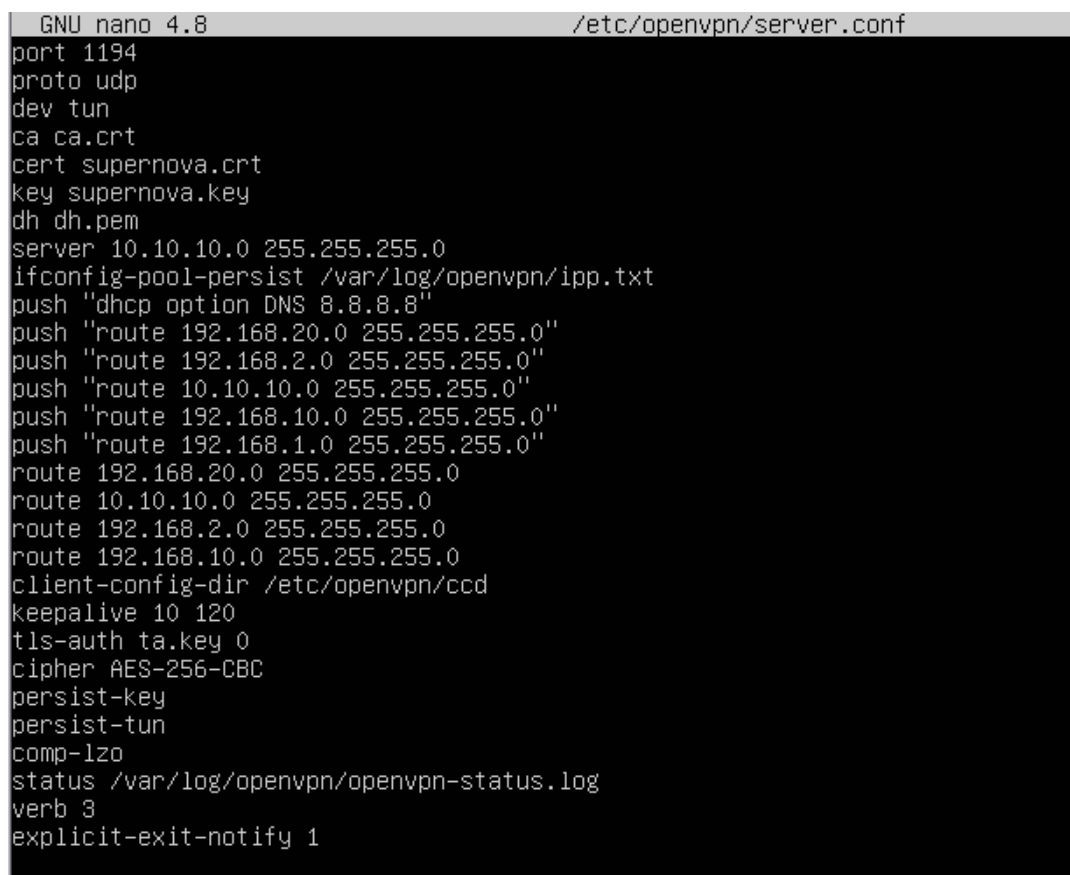
*Рисунок 16 – Перенос ключей и сертификатов.*

Теперь перейдем к созданию конфигурации сервера. Для этого с помощью команды “nano” в директории “/etc/openvpn” создадим файл “server.conf” (Рис.17).

В поле “port” указывается порт, на котором будет работать OpenVPN. В поле “proto” указывается транспортный протокол UDP или TCP. В поле dev тип интерфейса. В поле “ca” указывается путь до публичного сертификата центра сертификации. В поле “cert” указывается публичный сертификат OpenVPN сервера. В поле “key” приватный ключ OpenVPN сервера. В поле “dh” прописывается путь до файла Диффи-Хелмана. В поле “server” прописывается используемая сеть для OpenVPN. Директива “ifconfig-pool-persist” необходима для того, чтобы одним и тем же клиентам выдавался один и тот же IP адрес, после этой директивы указывается путь до расположения файла, в котором записываются соответствия, выданного первый раз IP адреса и клиента. Директива “push” используется для передачи сетевой конфигурации клиентам. Директива “route” позволяет серверу OpenVPN маршрутизировать указанные сети. Директива “client-config-dir” нужна чтобы указывать дополнительные параметры клиентов (к примеру, в данной работе создан файл с именем клиента в директории /etc/openvpn/ccd и в нем указаны сети из филиала – это необходимо для работы site-to-site VPN). Директива

“keepalive” указывает время до проверки доступности клиентов. В поле “tls-auth” указывается путь до ключа HMAC и цифра либо 0, либо 1 где 0 говорит, что это сервер, а 1 что это клиент. В поле “cipher” указывается набор алгоритмов для шифрования трафика. Директивы “persist-key” и “persist-tun” необходимы чтобы сервер не читал заново сертификаты при разрыве соединения. Директива “comp-lzo” нужна для того, чтобы сервер сжимал трафик перед отправлением (ускоряет работу OpenVPN). В директиву “status” указывается путь до файла, где будут храниться логи OpenVPN. Директива “verb” нужна чтобы указать глубину логирования (принцип схож с syslog).

Таким образом поля являются обязательными для заполнения без них не будет работать сервер, а директивы являются дополнительными настройками для OpenVPN.



```
GNU nano 4.8 /etc/openvpn/server.conf
port 1194
proto udp
dev tun
ca ca.crt
cert supernova.crt
key supernova.key
dh dh.pem
server 10.10.10.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push "dhcp option DNS 8.8.8.8"
push "route 192.168.20.0 255.255.255.0"
push "route 192.168.2.0 255.255.255.0"
push "route 10.10.10.0 255.255.255.0"
push "route 192.168.10.0 255.255.255.0"
push "route 192.168.1.0 255.255.255.0"
route 192.168.20.0 255.255.255.0
route 10.10.10.0 255.255.255.0
route 192.168.2.0 255.255.255.0
route 192.168.10.0 255.255.255.0
client-config-dir /etc/openvpn/ccd
keepalive 10 120
tls-auth ta.key 0
cipher AES-256-CBC
persist-key
persist-tun
comp-lzo
status /var/log/openvpn/openvpn-status.log
verb 3
explicit-exit-notify 1
```

*Рисунок 17 – Конфигурация сервера.*

Перейдём в директорию “/etc/openvpn/ccd” и создаём файл с названием клиента (Piter) в котором прописываем сети филиала, которые будет маршрутизироваться (Рис.18).

```
GNU nano 4.8 /etc/openvpn/ccd/Piter
iroute 192.168.20.0 255.255.255.0
iroute 192.168.2.0 255.255.255.0
```

*Рисунок 18 – Маршрутизируемые сети филиала.*

Теперь необходимо включить маршрутизацию на сервере OpenVPN (Рис.19).

```
root@supernova:/etc/openvpn# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@supernova:/etc/openvpn# _
```

*Рисунок 19 – Включение маршрутизации.*

Поднимем OpenVPN с помощью systemd, для этого введем команду “systemctl start openvpn@server”. Где “server” название конфигурационного файла. После проверим статус сервера командой “systemctl status openvpn@server” (Рис.20).

```
root@supernova:~# systemctl start openvpn@server
root@supernova:~# systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled-runtime; vendor preset: enabled)
   Active: active (running) since Sun 2021-05-09 13:20:30 UTC; 8min ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 653 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 469)
    Memory: 2.3M
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─653 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --

May 09 13:20:30 supernova ovpn-server[653]: Socket Buffers: R=[212992->212992] S=[212992->212992]
May 09 13:20:30 supernova ovpn-server[653]: UDPv4 link local (bound): [AF_INET][undef]:1194
May 09 13:20:30 supernova ovpn-server[653]: UDPv4 link remote: [AF_UNSPEC]
May 09 13:20:30 supernova ovpn-server[653]: MULTI: multi_init called, r=256 v=256
May 09 13:20:30 supernova ovpn-server[653]: IFCONFIG POOL: base=10.10.10.4 size=62, ipv6=0
May 09 13:20:30 supernova ovpn-server[653]: ifconfig_pool_read(), in='Piter,10.10.10.4', TODO: IPv6
May 09 13:20:30 supernova ovpn-server[653]: succeeded -> ifconfig_pool_set()
May 09 13:20:30 supernova ovpn-server[653]: IFCONFIG POOL LIST
May 09 13:20:30 supernova ovpn-server[653]: Piter,10.10.10.4
May 09 13:20:30 supernova ovpn-server[653]: Initialization Sequence Completed
lines 1-23/23 (END)
```

*Рисунок 20 – Запуск сервера.*

## 2.2 Конфигурирование и подключение клиентов.

На данном этапе все готово для генерации ключей и сертификатов клиента. Перейдем в директорию центра сертификации и запустим команду “./easyrsa build-client-full Piter nopass”. Этой командой мы обращаемся к центру сертификации для генерации ключей и сертификатов для клиента Piter. Параметр “nopass” опускает создание пароля для ввода клиентом перед подключением. После генерации выводятся пути, куда сохранены сертификаты и ключи (Рис.21).

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-client-full Piter nopass
Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/Piter.key.kTINjdusX5'
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'Piter'
Certificate is to be certified until Apr 20 10:34:53 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
```

*Рисунок 21 создание ключей и сертификатов клиента.*

Создадим в директории “/etc/openvpn/client” директорию “Peter” в которую поместим конфигурацию и ключи данного клиента (Рис.22).

```
root@supernova:~# mkdir /etc/openvpn/client/Piter
```

*Рисунок 22 – Создание директории для клиента Piter.*

Переносим нужные файлы для подключения к серверу OpenVPN и создаем конфигурацию клиента в созданной ранее директории (Рис.23).

```
root@supernova:/etc/openvpn/client# ls
Piter.conf  Piter.crt  Piter.key  ca.crt  ta.key
```

*Рисунок 23 – Необходимые файлы.*

Теперь необходимо создать конфигурацию клиента. Для этого командой “nano Piter.conf” создадим файл конфигурации (Рис.24).

Первой директивой “client” мы указываем протоколу OpenVPN что дальше идет конфигурация именно клиента. После указываем тип интерфейса “tun”. Протокол UDP.

Далее указываем адрес сервера OpenVPN, а именно сброшенный в локальную сеть адрес 75.87.23.10 с портом 1194 (Раздел 2.1). Дальше конфигурация в большей степени аналогично конфигурации сервера.

```
GNU nano 4.8 Piter.conf
client
dev tun
proto udp
remote 75.87.23.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert Piter.crt
key Piter.key
remote-cert-tls server
comp-lzo
tls-auth ta.key 1
cipher AES-256-CBC
verb 3
```

*Рисунок 24 – конфигурация клиента.*

С клиента филиала подключимся по sftp к серверу OpenVPN для того, чтобы скачать конфигурации (Рис.25).

```
root@supernova:/home/erik# sftp erik@75.87.23.10
The authenticity of host '75.87.23.10 (75.87.23.10)' can't be established.
ECDSA key fingerprint is SHA256:RHS5Q4iAMrrn0ByjI7WjdtUvtrTNGsWfUpTddjiztyQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '75.87.23.10' (ECDSA) to the list of known hosts.
erik@75.87.23.10's password:
Connected to 75.87.23.10.
sftp> _
```

*Рисунок 25 – Подключение по sftp.*

После скачиваем необходимую “папку” с конфигурацией клиента с помощью команды “get -r Piter/” (Рис.26).

```
sftp> get -r Piter/
Fetching /etc/openvpn/client/Piter/ to Piter
Retrieving /etc/openvpn/client/Piter
/etc/openvpn/client/Piter/Piter.crt          100% 4479   50.9KB/s   00:00
/etc/openvpn/client/Piter/ta.key             100% 636    7.2KB/s    00:00
/etc/openvpn/client/Piter/Piter.key           100% 1704   19.3KB/s   00:00
/etc/openvpn/client/Piter/Piter.conf          100% 218    2.5KB/s    00:00
/etc/openvpn/client/Piter/ca.crt              100% 1196   13.6KB/s   00:00
sftp> _
```

*Рисунок 26 – Скачивание файлов по sftp.*

По умолчанию скачанные файлы находятся в директории домашнего пользователя (не root) поэтому перенесем необходимые файлы в директорию “/etc/openvpn/client” (Рис.27).

```
root@supernova:/etc/openvpn/client# cp -R /home/erik/Piter/. /etc/openvpn/client/
root@supernova:/etc/openvpn/client# ls
Piter.conf  Piter.crt  Piter.key  ca.crt  ta.key
root@supernova:/etc/openvpn/client#
```

*Рисунок 27 перенос файлов клиента в директорию клиента OpenVPN.*

Теперь все готово для запуска клиента филиала OpenVPN. Запускаем OpenVPN клиент и проверяем статус (Рис.28).

```
root@supernova:/etc/openvpn/client# systemctl start openvpn-client@Piter
root@supernova:/etc/openvpn/client# systemctl status openvpn-client@Piter
• openvpn-client@Piter.service - OpenVPN tunnel for Piter
   Loaded: loaded (/lib/systemd/system/openvpn-client@.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-05-06 10:58:50 UTC; 7s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 1179 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 469)
    Memory: 3.3M
   CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@Piter.service
           └─1179 /usr/sbin/openvpn --suppress-timestamps --nobind --config Piter.conf

May 06 10:58:51 supernova openvpn[1179]: Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with
May 06 10:58:51 supernova openvpn[1179]: Incoming Data Channel: Cipher 'AES-256-GCM' initialized with
May 06 10:58:51 supernova openvpn[1179]: ROUTE_GATEWAY 192.168.20.1/255.255.255.0 IFACE=enp0s3 HWADDR=
May 06 10:58:51 supernova openvpn[1179]: TUN/TAP device tun0 opened
May 06 10:58:51 supernova openvpn[1179]: TUN/TAP TX queue length set to 100
May 06 10:58:51 supernova openvpn[1179]: /sbin/ip link set dev tun0 up mtu 1500
May 06 10:58:51 supernova openvpn[1179]: /sbin/ip addr add dev tun0 local 10.10.10.6 peer 10.10.10.5
May 06 10:58:51 supernova openvpn[1179]: /sbin/ip route add 10.10.10.1/32 via 10.10.10.5
May 06 10:58:51 supernova openvpn[1179]: WARNING: this configuration may cache passwords in memory &
May 06 10:58:51 supernova openvpn[1179]: Initialization Sequence Completed
lines 1-23/23 (END)
```

*Рисунок 28 – Запуск клиента OpenVPN.*

Поскольку клиент OpenVPN также в данном случае является маршрутизатором то включим ipv4 маршрутизацию (Рис.29).

```
root@supernova:/etc/openvpn# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@supernova:/etc/openvpn# _
```

*Рисунок 29 – Включение маршрутизации.*

Теперь создадим сертификаты и конфигурацию для удаленного сотрудника (Рис.30). Для разнообразия выпустим сертификат клиенту под паролем, чтобы перед подключением к серверу клиент вводил пароль.

```
root@supernova:/etc/openvpn/easy-rsa# ./easyrsa build-client-full ErikCLI
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/ErikCLI.key.Nbgwa9z2Mx'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'ErikCLI'
Certificate is to be certified until Apr 23 14:41:13 2024 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
root@supernova:/etc/openvpn/easy-rsa# _
```

*Рисунок 30 – Создание ключей и сертификатов клиента.*

Создадим директорию “ErikCLI” в директории “/etc/openvpn/client” (Рис.31).

```
root@supernova:/etc/openvpn/easy-rsa# cd ..
root@supernova:/etc/openvpn# cd client/
root@supernova:/etc/openvpn/client# mkdir ErikCLI
```

*Рисунок 31 – Создание директории.*

Переносим сертификаты и ключи в созданную ранее директорию (Рис.32).

```
root@supernova:/etc/openvpn/client# cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/client/ErikCLI
root@supernova:/etc/openvpn/client# cp /etc/openvpn/easy-rsa/pki/ta.key /etc/openvpn/client/ErikCLI
root@supernova:/etc/openvpn/client# cp /etc/openvpn/easy-rsa/pki/issued/ErikCLI.crt /etc/openvpn/client/ErikCLI
root@supernova:/etc/openvpn/client# cp /etc/openvpn/easy-rsa/pki/private/ErikCLI.key /etc/openvpn/client/ErikCLI
```

*Рисунок 32 – Перенос ключей и сертификатов.*



Создание конфигурации клиента для удаленного сотрудника (Рис.33).

```
GNU nano 4.8 ErikCLI.conf
client
dev tun
proto udp
remote 75.87.23.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert ErikCLI.crt
key ErikCLI.key
remote-cert-tls server
comp-lzo
tls-auth ta.key 1
cipher AES-256-CBC
verb 3
```

Рисунок 33 – Конфигурация клиента удаленного сотрудника.

Подключемся с клиента по ssh к серверу OpenVPN с помощью программы Bitvise SSH Client (Рис.34). Для того чтобы скачать конфигурацию клиента.

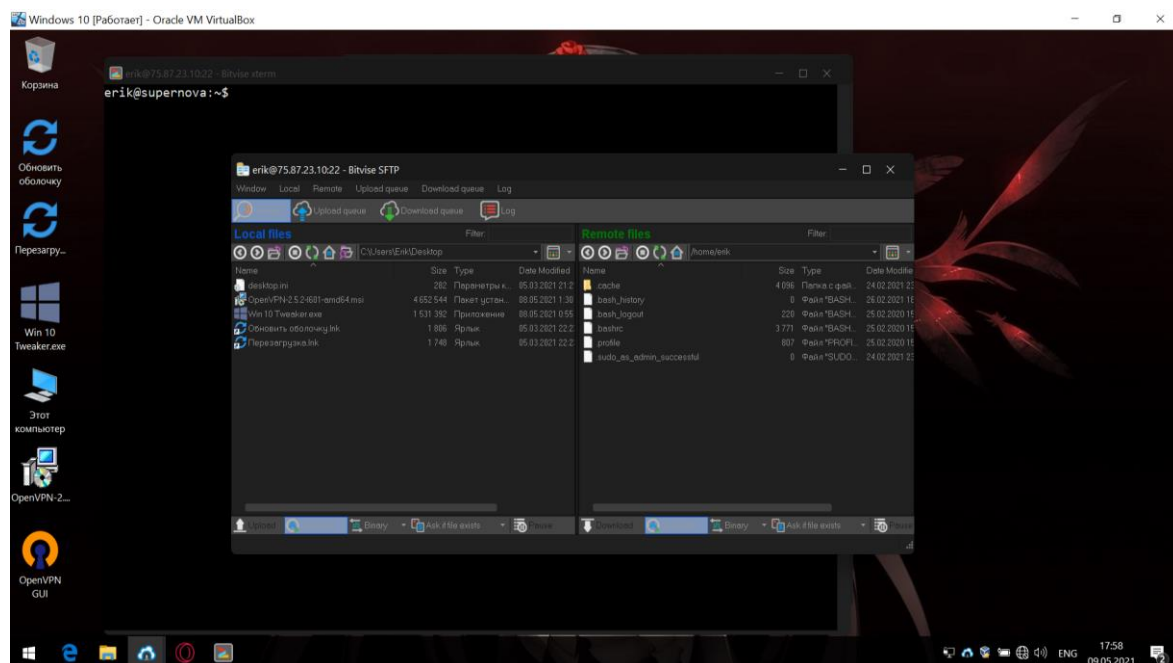


Рисунок 34 – Подключение по ssh.

Переходим в sftp клиенте в директорию “/etc/openvpn/client” и переносим папку ErikCLI на рабочий стол удаленного сотрудника (Рис.35).

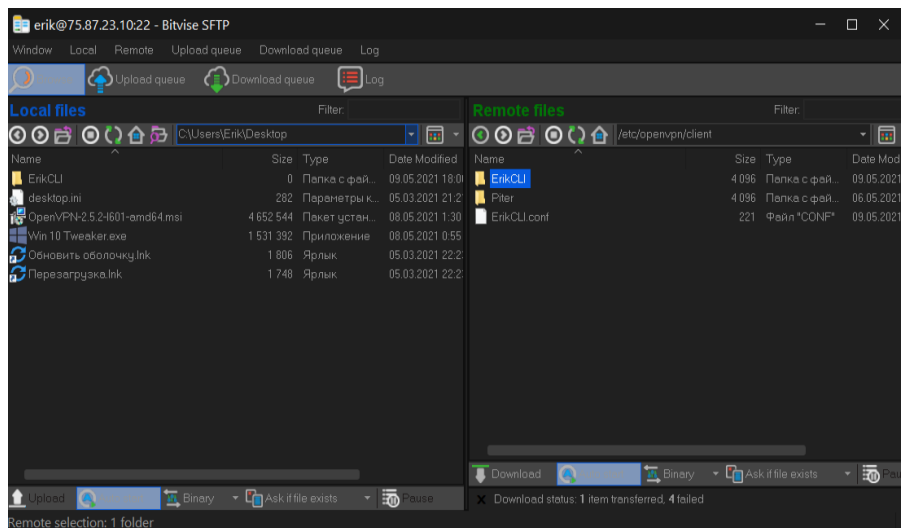


Рисунок 35 – Перенос папки с конфигурации клиента.

Для того чтобы установить OpenVPN соединение на Windows 10 необходимо загрузить с официального сайта OpenVPN программу OpenVPN GUI. После необходимо импортировать конфигурацию в программу (Рис.36). Перед тем как импортировать конфигурацию необходимо изменить расширение файла конфигурации на “.ovpn”.

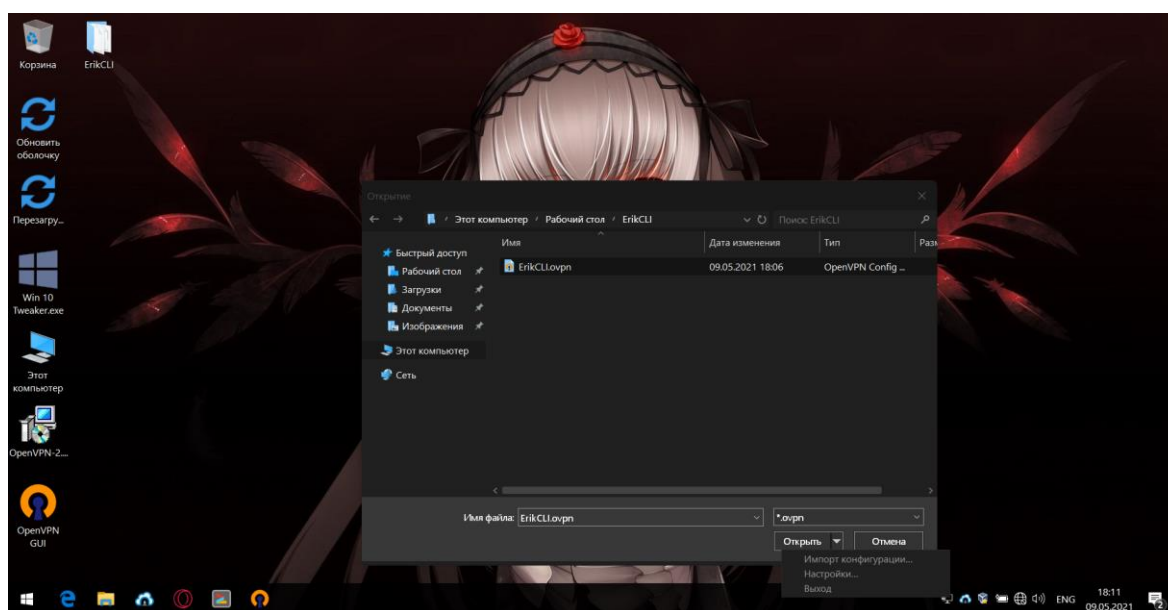


Рисунок 36 – Импорт Конфигурации.

Теперь можно произвести инициализацию подключения (Рис.37).  
Вводим пароль от ключа указанный при генерации (Рис.30)

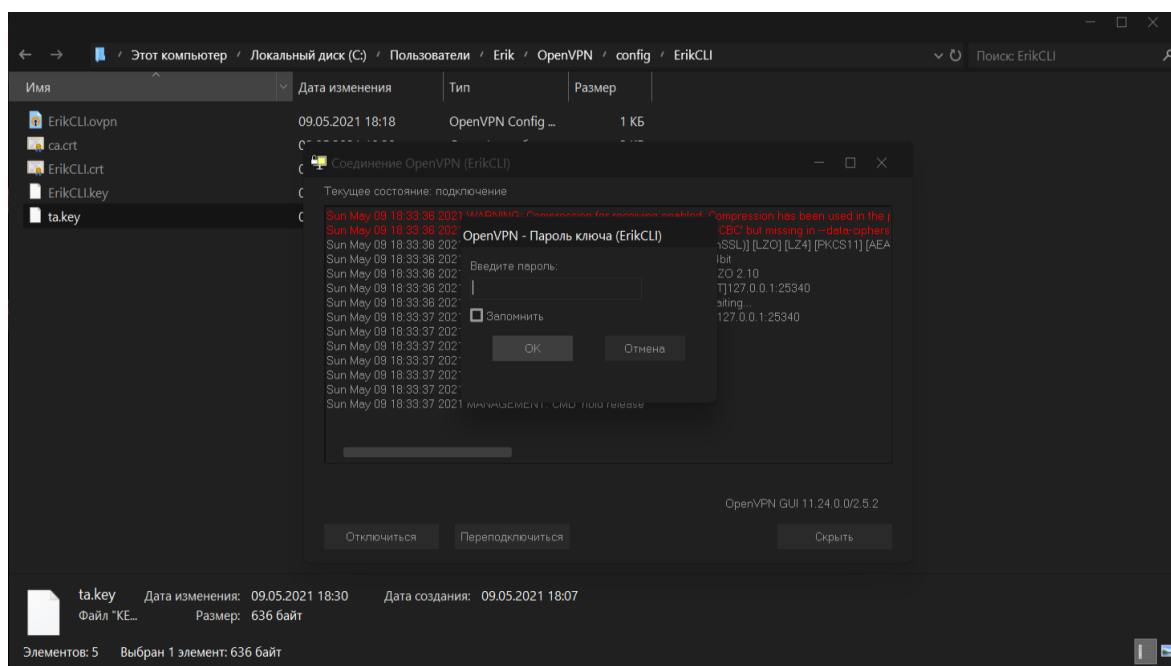


Рисунок 37 – Инициализация подключения и ввод пароля.

Посмотрим созданный интерфейс введя команду “ipconfig” в командной строке Windows (Рис.38).

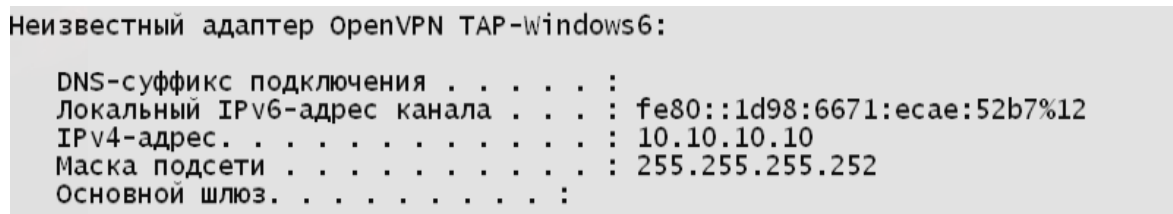


Рисунок 38 – Интерфейс OpenVPN.

## 2.3 Проверка работоспособности site-to-site OpenVPN.

Для того чтобы проверить работоспособность созданной сети OpenVPN создадим сетевую папку “Share” на Windows сервере главного офиса (DC1) (Рис.39).

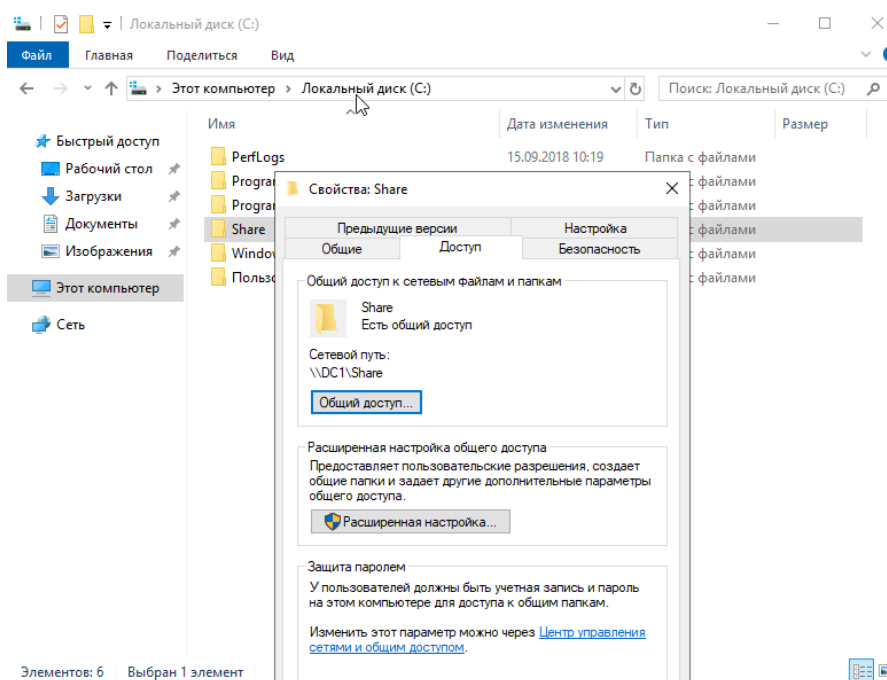


Рисунок 39 – Сетевая папка на DC1.

Перейдем на DC2 и произведем подключение сетевой папки расположенной на DC1 по локальному адресу 192.168.1.2 (Рис.40).

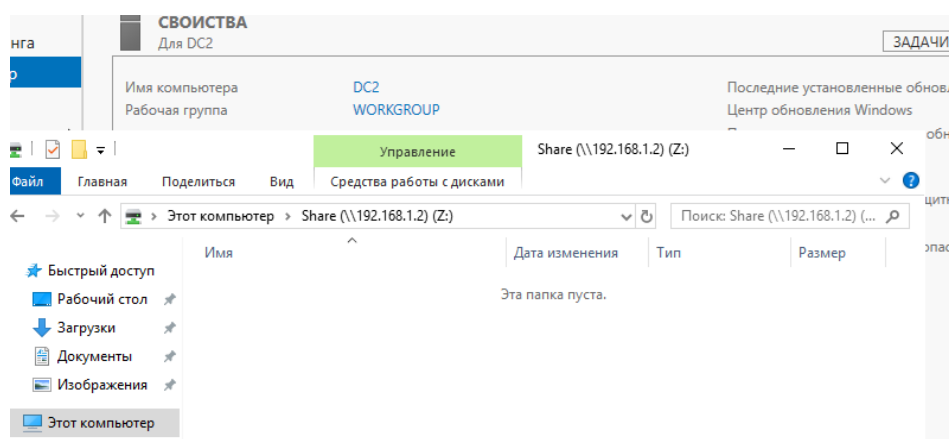


Рисунок 40 – подключение сетевой папки как диск Z.

Создадим еще одну сетевую папку “Share2”, но на сервере DC2 который находится в сети филиала (Рис.41).

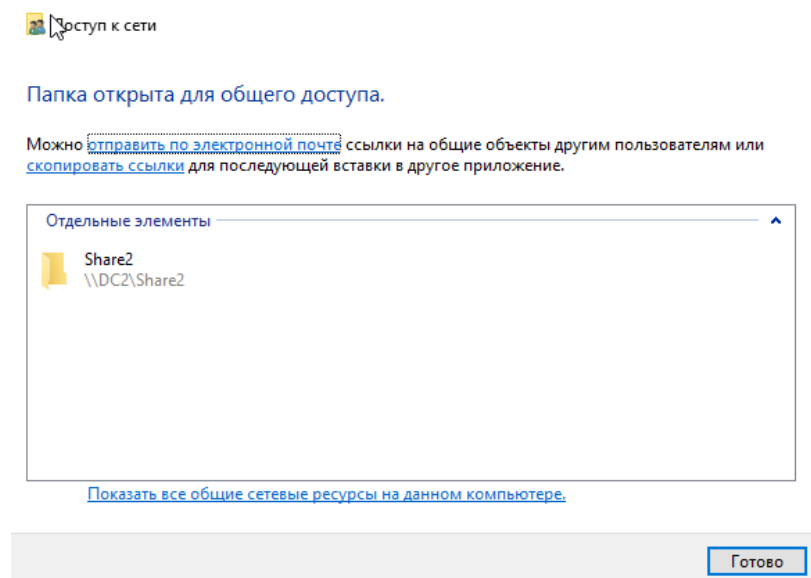


Рисунок 41 – Создание сетевой папки.

Перейдем на компьютер удаленного сотрудника и подключим обе сетевые папки, находящиеся на сервере главного офиса и сервере филиала (Рис.42)

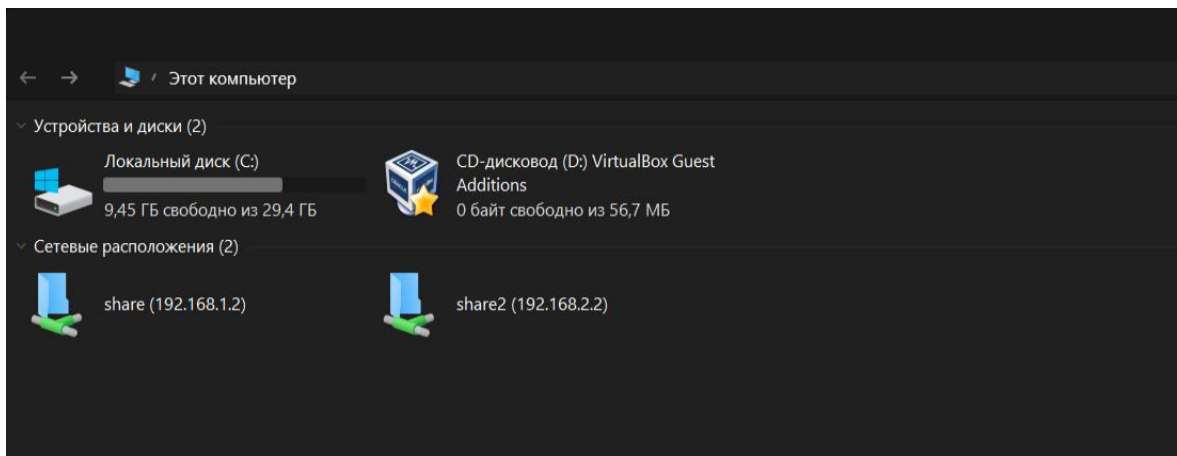


Рисунок 42 – Подключенные сетевые папки.

По итогу всё работает и сетевые папки подключаются что говорит о том, что у хостов в разных локальных сетях есть доступ к друг другу.

## ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе была рассмотрена тема: “Развертывание технологии Open VPN в корпоративной сети.” Виртуальные частные сети всё чаще используются в современном мире для решения таких задач как: обход блокировок провайдером сайтов, объединение локальных сетей филиалов офиса, предоставление удаленного доступа к локальной сети. Особенно чаще стали использовать технологию VPN для подключения удаленных сотрудников к корпоративной сети из-за пандемии COVID-19.

Существует множество разных VPN протоколов, но в данной работе рассматривался такой протокол как Open VPN. Один из самых популярных и надежных протоколов с открытым исходным кодом и поддержкой большого количества вариантов шифрования.

В теоретической части ВКР были рассмотрены следующие темы:

- Основные термины связанные с VPN. Принцип работы туннелирования и VPN, в частности.
- Протокол Open VPN. Сертификаты и центр сертификации. Варианты развертывание технологии Open VPN в корпоративной сети.
- Актуальность протокола Open VPN в сравнение с новым протоколом WireGuard.

В практической части была выполнена работа по развертыванию протокола Open VPN в корпоративной сети для объединения локальных сетей офиса и недавно открытого филиала, также был настроен удаленный доступ к корпоративной сети для удалённых сотрудников в рамках проекта в программе GNS3.

Для настройки Open VPN использовались два Ubuntu 20.04 сервера. В главном офисе был настроен Open VPN сервер, а в филиале клиент, также был настроен клиент для ПК с Windows 10 эмитирующий удаленного сотрудника. Поскольку сервера находятся за маршрутизаторами необходимо было настроить технологию NAT для проброса портов (в настройки использовался порт UDP 1194). Также на маршрутизаторах были прописаны статические маршруты до Ubuntu серверов для указания доступных сетей филиалов.

Поскольку сертификаты клиента создаются на сервере с Open VPN то для их переноса на клиента в ВКР использовался протокол SFTP. Для того чтобы по SFTP подключиться с клиента к серверу был настроен NAT для проброса 22 TCP порта до сервера Open VPN.

Для проверки работоспособности VPN туннеля были созданы на виртуальных машинах Windows server 2019 сетевые папки. После сетевые папки были подключены на серверах. Также была показана возможность подключение этих сетевых папок к клиенту.

Можно с уверенностью сказать, что задача, поставленная в данной выпускной квалификационной работе, была выполнена.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Авторы: Максимов Николай Вениаминович Попов Игорь Иванович Компьютерные сети (2019) — Текст: электронный — URL: <https://znanium.com/catalog/document?pid=792685>
2. Авторы: Кузин Александр Владимирович Кузин Дмитрий Александрович Компьютерные сети (2019) — Текст: электронный — URL: <https://znanium.com/catalog/document?pid=938938>
3. Компьютерные сети: расширенный начальный курс Авторы: А. А. Букатов, С. А. Гуда (2019) — Текст: электронный — URL: <https://www.litres.ru/aleksandr-pyhalov-un/komputernye-seti-rasshirennyy-nachalnyy-kurs-48613245/>
4. Олифер, Олифер: Компьютерные сети. Принципы, технологии, протоколы. Юбилейное издание (2020) — Текст: электронный — URL: <https://www.labirint.ru/books/737421/>
5. Олифер, Олифер: Компьютерные сети. Принципы, технологии, протоколы. (2019) — Текст: электронный — URL: <https://www.labirint.ru/books/511422/>
6. С. Грингард «Интернет вещей. «Будущее уже здесь» (2019)) — Текст: электронный — URL: <https://www.alpinabook.ru/catalog/book-75330/>
7. Компьютерные сети: Принципы, технологии, протоколы (2018)) — Текст: электронный — URL: [https://www.bsuir.by/m/12\\_100229\\_1\\_85460.pdf](https://www.bsuir.by/m/12_100229_1_85460.pdf)
8. Интернет изнутри. Экосистема глобальной сети. (2019)) — Текст: электронный — URL: <https://www.ozon.ru/context/detail/id/33354294/>



9. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-105: маршрутизация и коммутация) — Текст: электронный — URL: <https://www.ozon.ru/context/detail/id/147417590/>
10. Компьютерные сети. Учебник (2019)) — Текст: электронный — URL: <https://www.chitai-gorod.ru/catalog/book/1168613/>
11. Linux в действии (2019). Автор - Дэвид Клинтон — Текст: электронный — URL: [http://it-ebooks.ru/publ/unix/linux\\_in\\_action/14-1-0-1204](http://it-ebooks.ru/publ/unix/linux_in_action/14-1-0-1204)
12. Keijser J.J., OpenVPN Книга рецептов - второе издание, 2019 Текст: электронный — URL: <https://www.packtpub.com/product/openvpn-cookbook-sec>
13. Is Wireguard faster than OpenVPN? We tested 114 VPN servers., 2021 — Текст: электронный — URL: <https://vladtalks.tech/vpn/is-wireguard-faster-than-o>
14. Reference manual for OpenVPN 2.4, 2021 — Текст: электронный — URL: <https://openvpn.net/community-resources/reference-man>
15. Настройка OpenVPN в Ubuntu 20.04, 2021 — Текст: электронный — URL: <https://losst.ru/nastrojka-openvpn-v-ubuntu>
16. Site-to-site VPN routing explained in detail, 2021 — Текст: электронный — URL: <https://openvpn.net/vpn-server-resources/site-to-site-routing-explained-in-detail/>
17. Неизвестный Open VPN. Знакомимся со скрытыми возможностями и настройками, 2021 — Текст: электронный — URL: <https://xakep.ru/2020/04/23/unknown-openvpn/>

18. Настройка трансляции сетевых адресов, 2021 — Текст: электронный — URL: [https://www.cisco.com/c/ru\\_ru/support/docs/ip/network-address-translation-nat/13772-12.html](https://www.cisco.com/c/ru_ru/support/docs/ip/network-address-translation-nat/13772-12.html)
19. Cisoc NAT, 2021 — Текст: электронный — URL: [http://xgu.ru/wiki/Cisco\\_NAT](http://xgu.ru/wiki/Cisco_NAT).
20. Документация GNS3, 2021 Текст: электронный — URL: <https://docs.gns3.com/docs/>