Создаем свой VPN-сервер. Пошаговая инструкция

Данная статья позволит вам самостоятельно создать собственный VPN-сервер и настроить VPN-соединения на ваших iPhone, iPad и Mac. Защитить и скрыть ваш трафик от злоумышленников, провайдера и спецслужб, а также получить доступ к заблокированным в стране ресурсам.

В январе 2019 года я путешествовал по Азии. Будучи в Гуанчжоу, я столкнулся с отсутствием доступа к Google, Facebook и ряду других популярных сервисов. Они в принципе не работают на территории Китая и заблокированы правительством.

Помимо недоступности сервисов в некоторых странах, в путешествиях вообще есть определенная проблема с нормальным доступом в Интернет. Не всегда удается купить местную SIM-карту, поэтому приходится подключаться к Wi-Fi-сетям в отелях, аэропортах и кафе. Фактически, мы ничего не знаем об этих сетях: кем они были созданы, насколько

надежны и не «слушает» ли кто-либо сейчас наш трафик.

То же самое происходит и в своей стране. Если вы беззаботно подключаетесь к Wi-Fi в кафе или к любым публичным сетям, то у меня для вас плохие новости. Вы и понятия не имеете, насколько безопасны эти сети, и, вполне возможно, прямо сейчас весь ваш трафик прослушивается злоумышленниками, которые сидят за соседним столиком. Отличный пример показан на видео.

> **Вывод: всегда, когда мы подключаемся к любой сети, которая не является лично нашей, вся наша сетевая активность находится под постоянной угрозой.**

Использование VPN-соединения — оптимальное решение сразу обеих обозначенных проблем: так мы получаем доступ к заблокированным в стране ресурсам и защищаем себя от прослушки трафика. Реализовать это решение можно двумя способами:

- Воспользоваться коммерческим VPN-сервисом, например, Nord VPN
- Создать свой VPN-сервер

Для меня использование коммерческого VPN было неприемлемым решением, и по ходу статьи я объясню, почему никому не советую пользоваться подобными сервисами. А поскольку я уже имел навык работы с Linux, я решил создать собственный VPN-

сервер: размять пальцы, освежить знания, прокачать скилл.

Однако, побродив по Интернету, я не нашел ни одной работающей инструкции: в большинстве из них содержались ошибки, и ни одна из них не работала на 100%. Либо же эти инструкции были заточены под слишком специфические кейсы использования, что не подойдет для обычного среднестатистического пользователя. В связи с этим, я решил написать свою инструкцию. Надеюсь, она поможет тем, у кого есть потребность в создании собственного VPN-сервера. На мой взгляд, сегодня она должна быть у любого пользователя, который хоть сколько-нибудь заботится о безопасности своих данных.

Данная инструкция написана для широкого круга читателей и может быть применена жителем любой страны, однако, предполагает хотя бы минимальные знания Linux на уровне начального пользователя, который умеет обращаться с базовыми командами консоли. Инструкция проверена и обкатана несколько раз, поэтому гарантировано сработает на чистой системе без каких-либо ошибок. Настройка займет от 15 минут, в зависимости от скорости вашей работы.

К сожалению, Windows и Android в данной статье не рассматриваются, поскольку Windows последний раз я пользовался лет 15 назад, а Android — 2 часа

совокупно за всю жизнь, поэтому не представляю, что и как сегодня работает в этих операционных системах. Не исключаю, что подходы, описанные здесь, сработают и для них. Инструкция в первую очередь создана для тех, кто пользуется iPhone, iPad и Mac.

Как работает VPN, и почему вам нужен собственный VPN-сервер

С помощью вашего обычного Интернет-соединения между вашим устройством и VPN-сервером устанавливается специальное соединение — VPN-туннель. Все передаваемые и получаемые данные в этом соединении шифруются. С этого момента вся ваша сетевая активность осуществляется через данный туннель, а не через основной канал провайдера, и вы пользуетесь Интернетом как бы с самого VPN-сервера, а не с вашего устройства.

Для вашего провайдера, администратора Wi-Fi-сети или же злоумышленников, которые сканируют трафик в Wi-Fi-сетях, вся ваша сетевая активность выглядит как одно единственное соединение к одному единственному IP-адресу. Это всё, что им доступно. Что именно происходит внутри этого соединения, они не смогут узнать, поскольку просто не смогут проникнуть «внутрь» этого соединения.

Да, можно взломать сам VPN-сервер и получить

доступ к вашему трафику уже на нем, но очевидно, что никто не будет этим заниматься. К тому же, взломать хорошо защищенный VPN-сервер — тот еще челлендж.

Преимущества VPN

Итак, давайте обозначим, что нам дает VPN-соединение и какие преимущества предлагает:

- VPN-соединение обеспечит безопасность при подключении к ненадежным (не лично нашим) Wi-Fi-сетям и особенно полезно в путешествиях. Ни администратор Wi-Fi-сети, ни злоумышленники, сканирующие трафик, не смогут понять на какие сайты вы ходите, какие данные передаете или получаете.
- VPN-соединение обеспечит доступ к ресурсам, заблокированным в вашей стране или стране, в которой вы сейчас находитесь. Поскольку VPN-сервер находится за пределами вашей страны, а вместе с ним виртуально и вы, вам становятся доступны любые ресурсы Интернета. Если конечно они не заблокированы в стране нахождения самого VPN-сервера.
- VPN-соединение позволяет скрыть трафик от Интернет-провайдера и, соответственно, от спецслужб вашей страны. В России, например, уже имеются прецеденты посадки людей в тюрьмы всего за один комментарий в

социальной сети, а согласно «закону Яровой» весь ваш трафик и сетевая активность записываются сроком на 1 год. Многим кажется, что законопослушному гражданину бояться вроде бы нечего, однако, на мой взгляд, это существенное упущение в персональной безопасности. Кто знает, как изменится ваша личная ситуация и ситуация в стране. Вполне возможно, что в один «прекрасный» день к вам придут и предъявят обвинения за посещение неугодного правительству сайта. В конце концов, зачем вам давать кому-либо больше информации о себе, если можно дать меньше?

Безусловно, не стоит воспринимать VPN как панацею от всего и вся. Не стоит также и думать, что с использованием VPN вы сможете начать заниматься хакерской деятельностью, ломать сети и воровать кредитные карточки, как это часто показывают в кинофильмах, например в Mr.Robot. Во-первых, такая деятельность в принципе незаконна и неоднозначна с моральной точки зрения. Во-вторых, вас очень быстро обнаружат, обратившись к хостеру, у которого размещен VPN-сервер, и последний сдаст ваши реальные координаты с потрохами. В-третьих, для этого существуют совершенно другие инструменты и подходы.

> **Поэтому созданием собственного VPN-сервера мы в первую очередь обеспечиваем защиту от злоумышленников и**

> **излишне милитаризованной системы государства, получая свободу пользования Интернетом.**

Недостатки VPN

У VPN существуют и недостатки:

- Некоторые сайты начнут загружаться на языке страны, в которой располагается ваш VPN-сервер. Однако, это легко исправить. Поскольку чаще всего мы приходим на какие-либо сайты через поиск в Google, достаточно один раз настроить его выдачу, и с этого момента вы будете направляться на сайты с нужной вам локализацией. В конце концов, всегда можно выбрать локализацию на самом сайте.
- То же самое касается и рекламы. Ее существенная часть начнет отображаться для страны, в которой находится ваш VPN-сервер. Какие-то площадки научились понимать, что на самом деле вы русскоговорящий пользователь и находитесь совсем не в Германии, однако, YouTube, например, по-прежнему этого не умеет и шпарит видео-рекламу на немецком даже при принудительной установке страны в настройках. Впрочем, иногда это даже интересно: можно посмотреть какие продукты и услуги сейчас актуальны в других странах.
- Некоторые сервисы, например, Амедиатека, блокируют доступ с нероссийских IP-адресов,

поэтому VPN при их использовании придется на время отключать. Опять же, есть хак: достаточно выключить VPN, запустить видео в сервисе и сразу же включить VPN обратно. Все будет работать.
- Снизится скорость загрузки сайтов. По замерам это действительно так, однако, на практике падение скорости для обычного пользования Интернетом настолько незначительно и неощутимо, что данным недостатком можно пренебречь.

О коммерческих VPN-сервисах: NordVPN, ExpressVPN, Cyberghost, и почему не стоит их использовать

В настоящее время существуют так называемые коммерческие VPN-сервисы — компании, которые предоставляют вам свои VPN-сервера за определенную месячную плату. Например, NordVPN, ExpressVPN, CyberGhost и прочие.

На мой взгляд, использование подобных сервисов, несмотря на их предназначение, напротив, **еще больше снижает безопасность ваших данных**. Всё просто: эти компании видят весь ваш трафик. Всё, что вы получаете и передаете, на какие сайты ходите, какими сервисами пользуетесь — абсолютно всё. Да, конечно, все они заявляют о том, что не хранят клиентские логи, однако, это невозможно проверить

на практике. Более того, **некоторые VPN-сервисы изначально были созданы именно для того, чтобы воровать ваши данные**, и, по-факту, это является их основной деятельностью и выгодой.

Нельзя не отметить и низкое качество услуг: большие пинги, существенные просадки в скорости соединения. Оно и понятно, ведь на одном сервере располагаются сотни, если не тысячи, таких же клиентов, как и вы.

> **Таким образом, использование коммерческих VPN-сервисов «для всех» абсурдно само по себе. Лучше уж тогда вообще не использовать никакой VPN, чем сливать свои данные непонятно кому.**

Иногда использование коммерческих VPN может быть оправдано, но исключительно как эпизодическая и экстренная мера. Например, вы попали в страну, где какой-то нужный вам ресурс заблокирован, своего VPN-сервера у вас нет, а доступ нужен срочно.

В какой стране поднять свой VPN-сервер

Выбор страны для своего VPN-сервера стоит осуществлять, исходя из следующих критериев:

- кратчайшее расстояние до вас: это обеспечит меньший пинг и потери в скорости соединения
- минимальное количество запретов на свободу

Интернета, доступность любых популярных сервисов

- наименьшая политическая напряженность между вашей страной и страной, где будет находиться VPN-сервер. В этом случае ваш трафик теперь уже с VPN-сервера, скорее всего, не будут читать спецслужбы другого государства. Но здесь палка о двух концах. Например, многие российские пользователи предпочитают располагать VPN-сервер в Великобритании именно из-за высокой напряженности между странами, ведь в случае чего, последняя ни за что не сдаст трафик российским спецслужбам. Поэтому, данный подход также может быть оправданным.

Для российских пользователей в целом подойдет любая страна Евросоюза, однако, практика показывает, что лучшим решением является Германия: отличные пинг и стабильность канала, незначительные потери в скорости и хорошая доступность любых мировых ресурсов.

Если исходить из принципа максимальной защищенности трафика именно от российских спецслужб, то лучшим решением будет сервер в Великобритании. На практике же, лондонские сервера не всегда хорошо показывают себя с точки зрения стабильности: случаются отвалы, так себе пинг и меньшая, по сравнению с Германией,

скорость. Возможно, автор здесь предвзят, но сколько бы я не работал ни с чем британским, всё почему-то всегда работает из рук вон плохо, начиная от автомобилей и заканчивая серверами. Складывается впечатление, что разгильдяи-британцы *ничего не могут сделать хорошо*, в то время как на немецкую машину и ее педантичность всегда можно положиться.

Безусловно, вы можете поднять VPN-сервер и в своей стране, но в этом случае утрачиваются преимущества №2 и №3. Создавая VPN-сервер в своей стране, вы на блюдечке предоставляете весь свой трафик спецслужбам вашей страны, поскольку сервер, расположенный в вашей стране, подчиняется юрисдикции именно вашей страны. И, установив контроль за вашим трафиком пусть не у провайдера, но у самого хостера, где работает ваш VPN-сервер, спецслужбы при желании легко достанут вас там. И конечно, по-прежнему не будут доступны заблокированные ресурсы.

Выбираем хостинг для своего VPN-сервера

Популярные хостеры

Для того, чтобы создать свой VPN-сервер, нам нужно арендовать виртуальный сервер (Virtual Private Server) у одного из хостинг-провайдеров. На него мы

установим Linux и затем настроим его.

Выбор хостера дело персональное, на форумах существует бесчисленное количество топиков в духе «где лучше взять виртуальный сервер для VPN». Из наиболее популярных на сегодняшний день глобальных хостинговых компаний можно выделить следующие:

- Amazon Web Services
- DigitalOcean
- Hetzner
- Vultr
- Bluehost
- Arubacloud

Для себя я выбрал Amazon Web Services (AWS). В основном, из-за известности бренда, большого количества доступных географических зон для размещения сервера и высокой стабильности. На самом деле, многие популярные интернет-сервисы работают на базе AWS, арендуя там сервера для своих нужд, например, Facebook.

На мой взгляд, вряд ли сегодня кто-то может глобально конкурировать с Amazon. Компания была пионером в облачных технологиях и, по сути, открыла эту отрасль. Сегодня AWS предоставляет множество решений для облачных вычислений на любой вкус и цвет, но нам с вами нужна обычная виртуальная

машина. Ее мы возьмем в одной из разработок AWS: Lightsail.

Lightsail — это упрощенное решение для создания виртуальных серверов, в отличие от своего старшего собрата EC2. Всё завернуто в очень простой интерфейс, в котором разберется даже новичок, поэтому для нашей цели — создания VPN-сервера, AWS Lightsail подходит лучше всего.

Вообще, вы можете арендовать сервер у любой компании. Данная инструкция не сильно зависит от площадки и сработает у любого хостера.

Сколько стоит

Использование VPN-сервера на базе AWS Lightsail будет обходиться вам в 3.5 доллара в месяц. За эти деньги вы получаете машину с 512 Мб оперативной памяти. Подобная конфигурация легко справляется с обработкой VPN-трафика трех устройств. Первый же месяц у AWS будет и вовсе бесплатным.

Почему Debian, а не Ubuntu

Поднимать свой VPN-сервер мы будем на основе операционной системы Linux Debian, а не Linux Ubuntu, которая довольно часто фигурирует в подобных инструкциях.

Лично я не люблю Ubuntu с самого ее появления из-

за нестабильности, требовательности к ресурсам и какой-то общей аляповатости. К тому же, Ubuntu изначально создавалась именно как пользовательская система, а не серверная. Debian же надежен и стабилен как слон. В моей компании мы используем Debian во всех интернет-проектах на протяжении последних 10 лет и никогда не имели с ней проблем, получая феноменальные стабильность и быстродействие. С Ubuntu же *вечно что-то происходит*.

Протоколы VPN-соединения: почему IPsec IKEv2, а не Open VPN

Сегодня существуют разные протоколы VPN-соединения, их детальный разбор выходит за рамки этой статьи. Среди них наиболее популярны IPsec IKEv2 и OpenVPN.

Оба протокола хороши и надежны, но мы будем использовать IKEv2, поскольку у OpenVPN, на мой взгляд, имеется существенный недостаток, который перекрывает его прочие достоинства. OpenVPN требует установки своего приложения, которое всегда должно быть запущено на устройствах, что, во-первых, неудобно в использовании, а во-вторых, дополнительно расходует батарею iPhone, iPad и, в меньшей степени, Mac. IKEv2 же «вшит» в iOS и macOS и является для них нативным, не требуя установки никакого дополнительного ПО.

В качестве серверной части мы будем использовать strongSwan — популярный VPN-сервер для Linux.

Готовые скрипты для развертывания VPN-сервера: Algo, Streisand

Сегодня существуют готовые решения для развертывания своего VPN-сервера на платформе Linux, например, скрипт Algo (для IKEv2) или Streisand (для OpenVPN), которые нужно просто скачать, распаковать и запустить на сервере. Данные скрипты сами установят и настроят все необходимые пакеты и на выходе предоставят для вас работающий VPN-сервер.

Streisand нам не подходит в принципе, поскольку заточен под OpenVPN. Что же касается Algo, то пробежавшись по диагонали, я увидел, что данный скрипт устанавливает много лишних пакетов, без которых вполне можно обойтись, а также каких-то подозрительных пакетов, неизвестно кем созданных и кем проверенных. Если кто-то глубоко изучал Algo и нашел в нем что-то мошенническое, дайте, пожалуйста, знать. Кроме того, Algo устанавливается только на Ubuntu, что нам, опять же, не подходит.

Таким образом, мы будем создавать свой VPN-сервер, используя следующие технологии:

- AWS Lightsail в качестве виртуального сервера
- IKEv2 как протокол VPN
- Linux Debian в качестве серверной ОС
- strongSwan в качестве VPN-сервера
- никаких готовых скриптов, всё настроим руками.

Итак, с теоретической частью покончено, приступаем к созданию своего VPN-сервера.

Инструкция по созданию собственного VPN-сервера на базе Linux Debian

Регистрируемся в Amazon AWS и подключаемся к серверу

Процесс регистрации в Amazon Web Services прост и вы пройдете его самостоятельно. После регистрации перейдите в Lightsail, выберите гео-зону в которой вы хотите поднять свой VPN-сервер. Создайте новый инстанс, выберите «OS Only» и операционную систему Debian 9.5:

Нам подойдет простейшая машина с 512 Мб оперативной памяти:

Некоторые сайты (например, Авито) необоснованно блокируют визиты с IP-адресов популярных хостеров, считая, что последние часто принимают участие в DDOS-атаках. Чтобы этого не происходило и чтобы мы не делили один IP-адрес с тысячами других

машин AWS, перейдем в «Networking» и выделим себе Static IP. Назначим его на созданный нами инстанс:

Осталось скачать приватный ключ для доступа по SSH. Он находится в разделе «Account > SSH keys»:

Скачайте его и загрузите в SSH-клиент. Я настраивал сервер, находясь среди лиан на Бали со своего iPad Pro (да-да, то еще мучение!), используя Termius. Вы же можете использовать любой SSH-клиент, например, встроенный в macOS Terminal.app. В данной инструкции мы будем использовать именно его. Начинаем — открываем Terminal.

В первую очередь, чтобы избежать известной проблемы (у Termius таковой нет) терминала с тем, что называется «локаль», давайте добавим несколько новых строк в локальный файл macOS .profile. Отредактируем его, используя текстовый редактор nano:

```
nano ~/.profile
```

И вставим туда следующие строки:

```
export LC_ALL=en_US.UTF-8 export LANG=en_US.UTF-8
```

Сохраним файл с помощью Ctrl+X, закроем терминал с помощью Cmd+Q и запустим его снова.

Теперь перенесем скачанный приватный ключ

Lightsail в директорию, где хранятся ключи от SSH:

```
mv ~/Downloads/YOUR_DOWNLOADED_KEY.pem ~/.ssh
```

Ограничим ключу права, иначе macOS не позволит его использовать:

```
cd ~/.ssh/ chmod 600 YOUR_DOWNLOADED_KEY.pem
```

И подключимся к нашей Lightsail-машине, вписав вместо YOUR_LIGHTSAIL_IP её внешний статический IP-адрес:

```
ssh -i YOUR_DOWNLOADED_KEY.pem admin@YOUR_LIGHTSAIL_IP
```

Нас приветствует радостная надпись:

Переходим к настройке Debian

Все манипуляции будем осуществлять из-под пользователя **root**. Поехали:

В первую очередь, обновим индекс пакетов в репозиториях, возможно, есть обновления:

```
apt-get update
```

А затем установим эти обновления:

```
apt-get upgrade
```

Установка strongSwan

Установим strongSwan:

```
apt-get install strongswan
```

К детальной настройке strongSwan мы вернемся чуть позже, а пока создадим сертификаты, чтобы наши устройства смогли подключиться по VPN.

Генерируем сертификаты доступа

Мы будем использовать самозаверенные сертификаты, поскольку VPN-сервером планируем пользоваться только мы. Для того чтобы создать сертификаты, нам потребуется пакет strongswan-pki. Установим его:

```
apt-get install strongswan-pki
```

Переходим к созданию сертификатов. В первую очередь нам нужно создать корневой сертификат, он же “CA” (Certificate Authority), который выпустит нам остальные сертификаты. Создадим его в файле ca.pem:

```
cd /etc/ipsec.d ipsec pki --gen --type rsa --size 4096 --outform pem > private/ca.pem ipsec pki --self --ca --lifetime 3650 --in private/ca.pem \ > --type rsa --digest sha256 \ > --dn "CN=YOUR_LIGHTSAIL_IP" \ > --outform pem > cacerts/ca.pem
```

Далее создадим сертификат для нашего VPN-сервера в файле debian.pem:

```
ipsec pki --gen --type rsa --size 4096 --outform pem > private/debian.pem ipsec pki --pub --in private/debian.pem --type rsa | > ipsec pki --issue --lifetime 3650 --digest sha256 \ > --cacert cacerts/ca.pem --cakey private/ca.pem \ > --dn
```

```
"CN=YOUR_LIGHTSAIL_IP" \ > --san YOUR_LIGHTSAIL_IP \ > --flag
serverAuth --outform pem > certs/debian.pem
```

А теперь создадим сертификат для наших устройств в файле me.pem:

```
ipsec pki --gen --type rsa --size 4096 --outform pem > private/me.pem
ipsec pki --pub --in private/me.pem --type rsa | > ipsec pki --issue --
lifetime 3650 --digest sha256 \ > --cacert cacerts/ca.pem --cakey
private/ca.pem \ > --dn "CN=me" --san me \ > --flag clientAuth \ > --
outform pem > certs/me.pem
```

Для надежности удалим файл ca.pem, он нам больше не потребуется:

```
rm /etc/ipsec.d/private/ca.pem
```

Создание сертификатов завершено.

Если сертификаты генерируются слишком долго

Если ваши сертификаты генерируются слишком долго, например, более пяти секунд, это может свидетельствовать о низком количестве энтропии. С подобной ситуацией в свое время я столкнулся у Hetzner на их облачных машинах. Энтропии было слишком мало, и создание сертификатов растягивалось на 40-50 минут, поэтому в конце концов я решил отказаться от их услуг.

Проверить количество энтропии можно запустив еще один сеанс в соседней вкладке:

```
cat /proc/sys/kernel/random/entropy_avail
```

Данная команда выведет количество энтропии на момент запроса. Чтобы мониторить энтропию в реальном времени, выполните команду:

```
watch -n 0.25 cat /proc/sys/kernel/random/entropy_avail
```

Если энтропии меньше 200, я бы порекомендовал сменить хостера. Или же установить пакет haveged, который *якобы генерирует энтропию*, однако учтите, что тогда вы действуете на свой страх и риск. У Amazon Lightsail с энтропией обычно всё в порядке, поэтому ключи создаются мгновенно.

Выйти из запроса можно с помощью Ctrl+Z.

Настроим сам strongSwan

Очистим дефолтный конфиг strongSwan командой:

```
> /etc/ipsec.conf
```

И создадим свой в текстовом редакторе nano:

```
nano /etc/ipsec.conf
```

Вставьте данный текст в него, заменив YOUR_LIGHTSAIL_IP на внешний IP-адрес машины в AWS Lightsail:

```
include /var/lib/strongswan/ipsec.conf.inc config setup uniqueids=never charondebug="ike 2, knl 2, cfg 2, net 2, esp 2, dmn 2, mgr 2" conn %default keyexchange=ikev2 ike=aes128gcm16-sha2_256-prfsha256-ecp256! esp=aes128gcm16-sha2_256-ecp256! fragmentation=yes rekey=no compress=yes dpdaction=clear left=%any leftauth=pubkey
```

```
leftsourceip=YOUR_LIGHTSAIL_IP leftid=YOUR_LIGHTSAIL_IP
leftcert=debian.pem leftsendcert=always leftsubnet=0.0.0.0/0 right=%any
rightauth=pubkey rightsourceip=10.10.10.0/24 rightdns=8.8.8.8,8.8.4.4
conn ikev2-pubkey auto=add
```

Внимание! strongSwan требователен к отступам в конфиге, поэтому удостоверьтесь, что параметры каждого раздела конфига отбиты через Tab, как это показано на примере, или хотя бы через один пробел, иначе strongSwan не запустится.

Сохраним файл с помощью Ctrl+X и пойдем дальше.

Добавим в файл ipsec.secrets, который является хранилищем ссылок на сертификаты и ключи аутентификации, указатель на наш сертификат сервера:

```
nano /etc/ipsec.secrets
include /var/lib/strongswan/ipsec.secrets.inc : RSA debian.pem
```

На этом настройка Strongswan завершена, можно рестартнуть службу:

```
ipsec restart
```

Если всё хорошо, то сервер запустится:

```
... Starting strongSwan 5.5.1 IPsec [starter]...
```

Если упадет в ошибку, то можно посмотреть, что именно произошло, почитав системный лог. Команда выведет 50 последних строк лога:

```
tail -n 50 > /var/log/syslog
```

Настроим сетевые параметры ядра

Теперь нам необходимо внести некоторые изменения в файл /etc/sysctl.conf.

```
nano /etc/sysctl.conf
```

Через Ctrl+W найдем в файле следующие переменные и внесем в них изменения:

```
#Раскомментируем данный параметр, чтобы включить переадресацию пакетов net.ipv4.ip_forward = 1 #Раскомментируем данный параметр, чтобы предотвратить MITM-атаки net.ipv4.conf.all.accept_redirects = 0 #Раскомментируем данный параметр, чтобы запретить отправку ICMP-редиректов net.ipv4.conf.all.send_redirects = 0 ... #В любом месте файла на новой строке добавим данный параметр, запретив поиск PMTU net.ipv4.ip_no_pmtu_disc = 1
```

Подгрузим новые значения:

Настройка сетевых параметров ядра завершена.

Настроим iptables

iptables — это утилита, которая управляет встроенным в Linux файрволом netfilter. Для того, чтобы сохранять правила iptables в файле и подгружать их при каждом запуске системы, установим пакет iptables-persistent:

```
apt-get install iptables-persistent
```

После установки нас спросят, сохранить ли текущие правила IPv4 и IPv6. Ответим «Нет», так как у нас новая система, и по сути нечего сохранять.

Перейдем к формированию правил iptables. На всякий пожарный, очистим все цепочки:

```
iptables -P INPUT ACCEPT iptables -P FORWARD ACCEPT iptables -F iptables -Z
```

Разрешим соединения по SSH на 22 порту, чтобы не потерять доступ к машине:

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Разрешим соединения на loopback-интерфейсе:

```
iptables -A INPUT -i lo -j ACCEPT
```

Теперь разрешим входящие IPSec-соединения на UDP-портах 500 и 4500:

```
iptables -A INPUT -p udp --dport 500 -j ACCEPT iptables -A INPUT -p udp --dport 4500 -j ACCEPT
```

Разрешим переадресацию ESP-трафика:

```
iptables -A FORWARD --match policy --pol ipsec --dir in --proto esp -s 10.10.10.0/24 -j ACCEPT iptables -A FORWARD --match policy --pol ipsec --dir out --proto esp -d 10.10.10.0/24 -j ACCEPT
```

Настроим маскирование трафика, так как наш VPN-сервер, по сути, выступает как шлюз между Интернетом и VPN-клиентами:

iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -m policy --pol ipsec --dir out -j ACCEPT iptables -t nat -A POSTROUTING -s 10.10.10.0/24 -o eth0 -j MASQUERADE

Настроим максимальный размер сегмента пакетов:

iptables -t mangle -A FORWARD --match policy --pol ipsec --dir in -s 10.10.10.0/24 -o eth0 -p tcp -m tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1361:1536 -j TCPMSS --set-mss 1360

Запретим все прочие соединения к серверу:

iptables -A INPUT -j DROP iptables -A FORWARD -j DROP

Сохраним правила, чтобы они загружались после каждой перезагрузки:

netfilter-persistent save netfilter-persistent reload

Настройка iptables завершена.

Перезагрузим машину:

И посмотрим работают ли правила iptables:

sudo su iptables -S

root@XX.XX.XX.XX:/home/admin# iptables -S -P INPUT ACCEPT -P FORWARD ACCEPT -P OUTPUT ACCEPT -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT -A INPUT -i lo -j ACCEPT -A INPUT -p udp -m udp --dport 500 -j ACCEPT -A INPUT -p udp -m udp --dport 4500 -j ACCEPT -A INPUT -j DROP -A FORWARD -s 10.10.10.0/24 -m policy --dir in --pol ipsec --proto esp -j ACCEPT -A FORWARD -d 10.10.10.0/24 -m policy --dir out --pol ipsec --proto esp -j ACCEPT -A FORWARD -j DROP

Да, всё работает.

Работает ли strongSwan:

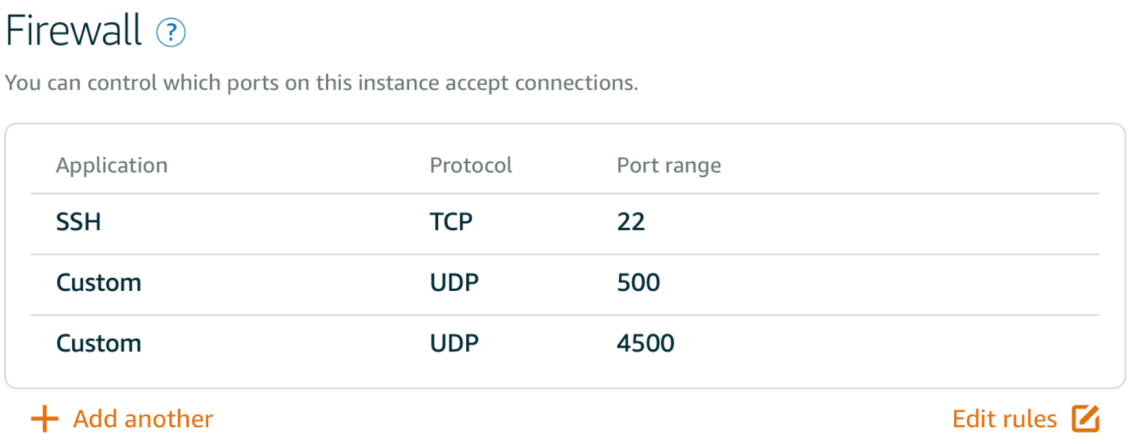
```
ipsec statusall
```

```
root@XX.XX.XX.XX:/home/admin# ipsec statusall Status of IKE charon daemon (strongSwan 5.5.1, Linux 4.9.0-8-amd64, x86_64): uptime: 71 seconds, since Jan 23 23:22:16 2019 ...
```

Да, всё работает.

Разрешаем соединения в файрволе Lightsail

AWS Lightsail использует также и свой файрвол для защиты виртуальных машин. Выберем наш инстанс, перейдем в «Networking» и разрешим соединения на UDP-портах 500 и 4500. По пути удалим ненужный нам 80-й порт:



Настройка файрвола Lightsail завершена.

Создаем .mobileconfig для iPhone, iPad и Mac

Мы будем использовать один VPN-профайл .mobileconfig для всех наших устройств: iPhone, iPad и Mac. Конфиг, который мы сделаем, устроен таким

образом, чтобы инициировать соединение “On Demand”. Это означает, что при попытке любой службы или приложения выйти в Интернет, VPN-соединение будет всегда устанавливаться принудительно и автоматически. Таким образом, удастся избежать ситуации, когда вы забыли установить VPN-соединение, например, после перезагрузки устройства, а трафик в итоге пошел через провайдера, что нам совсем не нужно.

Скачаем скрипт, который сгенерирует для нас данный конфиг:

```
wget https://gist.githubusercontent.com/borisovonline/955b7c583c049464c878bbe43329a521/raw/966e8a1b0a413f794280aba147b7cea0661f77a8/mobileconfig.sh
```

Для того, чтобы скрипт отработал, нам потребуется пакет zsh, установим его:

```
apt-get install zsh
```

Отредактируем название сервера по вкусу, а также пропишем внешний IP-адрес машины Lightsail, который мы указывали при создании сертификатов:

```
nano mobileconfig.sh
... SERVER="AWS Frankfurt" FQDN="YOUR_LIGHTSAIL_IP" ...
```

Запустим скрипт и на выходе получим готовый файл iphone.mobileconfig:

```
chmod u+x mobileconfig.sh ./mobileconfig.sh > iphone.mobileconfig
```

Заберите этот файл с сервера, подключившись с помощью любого SFTP-клиента, например, Transmit или Cyberduck, и отправьте его на все ваши устройства через Airdrop. Подтвердите на устройствах установку конфигурации.

Готово! Соединения с VPN-сервером установятся автоматически.

Приберемся за собой:

```
rm mobileconfig.sh rm iphone.mobileconfig
```

Прокачаем безопасность SSH (необязательный пункт)

Наш VPN-сервер уже работает и неплохо защищен, однако, я предлагаю еще немного прокачать безопасность SSH.

Для того, чтобы ботнеты не пытались пробиться к нам по SSH через дефолтный порт, перебирая пароли, и не оставляли в логах кучу мусора, изменим его на какой-нибудь другой, а также внесем ряд прочих изменений в конфигурацию SSH.

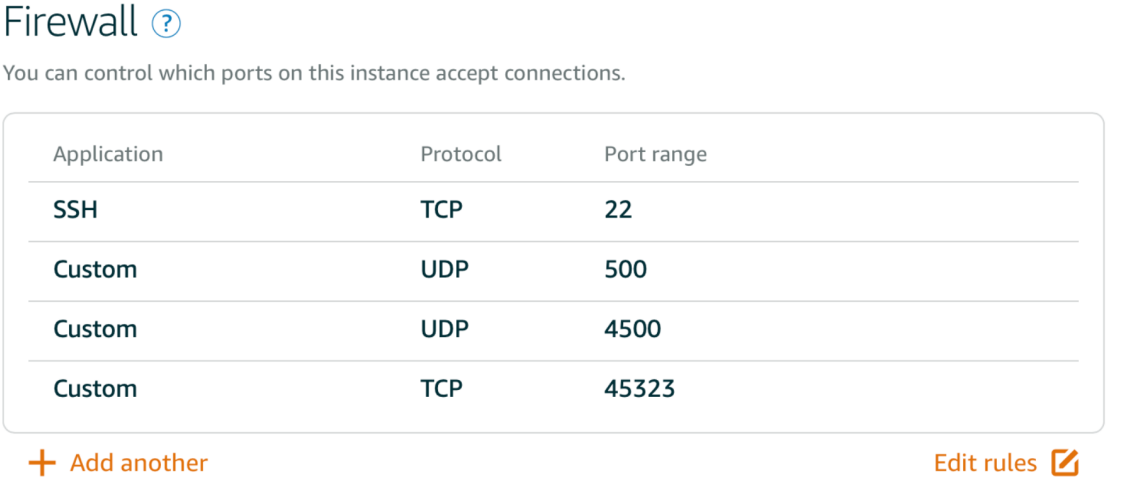
Вы можете выбрать любой порт по вкусу, начиная с 1024, однако, я рекомендую поискать такой порт, который не был замечен в использовании вирусами, троянами, а также не используется какими-либо

известными сервисами, программным обеспечением или производителями оборудования. Найдите себе такой «чистый» порт на SpeedGuide или adminsubnet.

В нашем примере мы будем использовать порт 45323.

Внимание! Не перезапускайте службы SSH и iptables и не перезагружайте машину, пока не пройдете данный раздел до конца, иначе вы потеряете доступ к машине!

Добавим новый 45323 TCP-порт в Lightsail:



Теперь настроим сам SSH:

nano /etc/ssh/sshd_config

#Раскомментируем и пропишем новый порт Port 45323 #Раскомментируем и запретим попытки соединения с пустым паролем PermitEmptyPasswords no #Раскомментируем и настроим автоматический разрыв соединения при бездействии через 360 секунд. Это будет полезно, если вы забыли, что у вас на экране осталась активная сессия и отошли от компьютера. Сервер автоматически разорвет SSH-соединение через 6 минут.

```
ClientAliveInterval 360 ClientAliveCountMax 0
```

Теперь обновим информацию в правилах iptables и заменим старый порт SSH на новый:

```
nano /etc/iptables/rules.v4
```

Заменим «22» в строке

```
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

на «45323»:

```
-A INPUT -p tcp -m tcp --dport 45323 -j ACCEPT
```

Сохраним файл через Ctrl+X и перезапустим машину:

После перезагрузки сервера проверьте соединение по SSH, добавив флаг «-p» и новый порт:

```
ssh -i YOUR_DOWNLOADED_KEY.pem admin@YOUR_LIGHTSAIL_IP -p 45323
```

Всё должно работать.

Не забудьте удалить старый порт 22 в настройках файрвола Lightsail.

Заключение

Итак, мы с нуля настроили свой собственный защищенный VPN-сервер и получили VPN-конфигурации для наших устройств. Теперь весь наш трафик зашифрован и недоступен ни провайдеру, ни

администратору Wi-Fi-сети, ни злоумышленникам, которые раньше могли его прослушивать. Теперь мы можем свободно подключаться к любым Wi-Fi сетям, не опасаясь за собственные данные. Кроме того, теперь нам доступны любые ресурсы, заблокированные на территории страны.

Вообще, вы можете создать себе сколько угодно VPN-серверов и переключаться между ними. Если вы часто путешествуете, то можно создать серверы в тех географических зонах, где вы чаще всего бываете: это обеспечит меньший пинг и более высокую скорость передачи данных. Для замера этих параметров удобно использовать приложение Speedtest.

Кроме того, бывает и так, что для определенной страны какой-либо ресурс недоступен по техническим причинам, в то время как из другой он работает исправно. В этом случае достаточно просто подключиться к другому VPN-серверу.

Делитесь инструкцией с друзьями, распространяйте в социальных сетях. Так, больше людей смогут защитить свои данные, а мир станет лучше.

Всем безопасного серфинга!

Дополнительные материалы к статье:

- Ассиметричное шифрование. Как это работает?

- Исчерпывающий мануал по iptables на русском
- Официальная документация strongSwan

#инструкции

Материал опубликован пользователем.
Нажмите кнопку «Написать», чтобы поделиться мнением или рассказать о своём проекте.

Написать