Настройка VPN сервера WireGuard в FreeBSD

VPN надёжно шифрует трафик, оберегая данные в публичных Wi-Fi, при сомнительных провайдерах или когда хочется больше анонимности и безопасности в сети. Сервер для VPN может быть куплен у провайдера VDS или можно использовать домашний сервер с внешним IP адресом. В статье разберём WireGuard — лёгкий, шустрый, открытый и безопасный VPN-протокол.

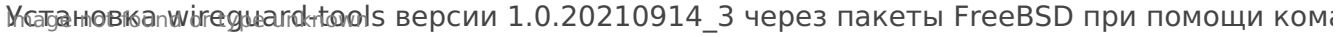
Оглавление статьи:

1. Установка WireGuard в FreeBSD

В FreeBSD уже давно WireGuard встроен в ядро и уже присутствует утилита wg, но для более легкой настройки удобнее использовать **wg-quick**.

Устанавливаем **wireguard-tools** через пакеты

```
pkg install wireguard-tools
```


Установка wireguard-tools версии 1.0.20210914_3 через пакеты FreeBSD при помощи ком

Или устанавливаем **wireguard-tools** из портов

```
cd /usr/ports/net/wireguard-tools/ && make install clean
```

2. Генерация ключей для сервера и клиента WireGuard

Для шифрования соединения необходимо сгенерировать приватный и публичный ключ для VPN сервера и клиента WireGuard. В официальном руководстве WireGuard показан пример, где ключи генерируются в каталог **/etc/wireguard/** каждый в отдельный файл. Эти файлы нужны только, для того чтобы перенести из них значения ключей в файлы конфигурации VPN сервера и клиента.

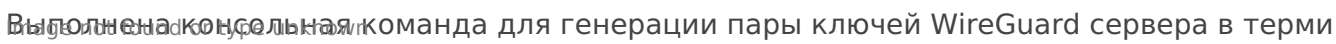
2.1. Генерация ключей сервера

Я накидал консольную команду которая выведет пару ключей на экран терминала и их можно будет скопировать в блокнот или использовать другое окно терминала. Для генерации приватного ключа используется команда `wg genkey`, для генерации публичного ключа используется команда `wg pubkey` и в качестве параметра для нее передается приватный ключ через временную переменную **WGKEY**.

Генерируем приватный и публичный ключ сервера:

```
WGKEY=$(wg genkey); echo "Server Private Key: $WGKEY"; echo $WGKEY | echo "Server Public Key: $(wg pubkey)"
```

Значение **Server Private Key** необходимо будет вставить в файл конфигурации сервера, а значение **Server Public Key** в файл конфигурации клиента.


Выполнена консольная команда для генерации пары ключей WireGuard сервера в терми

Генерировать ключи можно на любом компьютере где уставлен WireGuard, необязательно на сервере, где будет работать VPN сервер.

2.2. Генерация ключей клиента

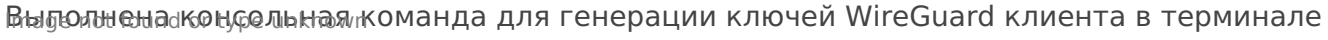
Консольная команда генерации ключей аналогичен предыдущей, но с добавлением генерации **Pre-Shared Key** через команду `wg genpsk` и заменой слова **Server** на **Client**

Ключ **Pre-Shared Key** используется для дополнительного шифрования внутри основного. Оно необходимо для защиты от расшифровки трафика инопланетными цивилизациями при помощи квантовых компьютеров или в будущем квантовым компьютером созданным человечеством.

Генерируем приватный и публичный ключ клиента:

```
WGKEY=$(wg genkey); echo "Client Private Key: $WGKEY"; echo $WGKEY | echo "Client Public Key: $(wg pubkey)"; echo "Client Pre-Shared Key: $(wg genpsk)"
```

Значения **Client Private Key** и **Client Pre-Shared Key** необходимо будет вставить в файл конфигурации клиента, а значение **Client Public Key** в файл конфигурации сервера.



Выполнена консольная команда для генерации ключей WireGuard клиента в терминале

Для каждого отдельного пользователя/устройства необходимо генерировать свои ключи.

3. Получение имени сетевой карты

Перед настройкой конфигурации сервера необходимо знать имя сетевой карты которая смотрит в интернет.

Выводим список всех сетевых интерфейсов и находим имя нужной сетевой карты

```
ifconfig
```

Или можно получить имя сетевой карты, которая использует шлюз по умолчанию командой

```
route -n get default | grep 'interface:' | awk '{print $2}'
```

```
em0
```

В моем случае команда вывела имя **em0** сетевого интерфейса, который получает интернет. Почти во всех случаях команда выведет имя нужной сетевой карты.

4. Настройка и конфигурация WireGuard сервера (wg0.conf)

Создаем файл конфигурации **wg0.conf** для VPN сервера WireGuard в каталоге **/usr/local/etc/wireguard/**

```
nano /usr/local/etc/wireguard/wg0.conf
```

Описание основных параметров конфигурационного файла **wg0.conf** (*строки для работы в сетях IPv6 закомментированы*)

```
# Файл конфигурации VPN сервера WireGuard
#
# Раздел "Interface" относится к настройкам сервера
[Interface]
# Приватный ключ сервера из значения "Server Private Key"
PrivateKey = gMtlDa6fcFLfKBRQiBJlaGzvVu9C0tGZrsQRg1DJMHU=
# IP адрес сервера и маска внутренней сети VPN
Address = 192.168.10.1/24
#Address = fd20:20:20::1/64
# Порт сервера
ListenPort = 51820

# Разделы "Peer" относится к настройкам клиентов
[Peer]
# Публичный ключ клиента из значения "Clinet Puplic Key"
PublicKey = MW5v76GNuyMXLyMmH6nDqQ98XdHNbdWluGB42J8SU3Y=
# Ключ PSK из значения "Clinet Pre-Shared Key"
PresharedKey = G1WVZwGj+bDzggvqx9jBTL6NH4dhlW5rCW0PCFHY2oY=
# IP адрес клиента (указание маски 32 обязательно для IPv4)
AllowedIPs = 192.168.10.2/32
#AllowedIPs = fd20:20:20::2/128
```

```
# Настройка для второго устройства
#[Peer]
#PublicKey = "Clinet Puplic Key"
#PresharedKey = "Clinet Pre-Shared Key"
#AllowedIPs = 192.168.10.3/32
```

Компактный файл конфигурации сервера выглядит так:

```
[Interface]
PrivateKey = gMtlDa6fcFLfKBRQiBJIaGzvVu9C0tGZrsQRg1DJMHU=
Address = 192.168.10.1/24
ListenPort = 51820

[Peer]
PublicKey = MW5v76GNuyMXLyMmH6nDqQ98XdHNbdWluGB42J8SU3Y=
PresharedKey = G1WVZwGj+bDzggvqx9jBTL6NH4dhlW5rCW0PCFHY2oY=
AllowedIPs = 192.168.10.2/32
```

Основные изменения которые необходимо внести в файл конфигурации:

- параметр **PrivateKey** необходимо изменить на свой приватный ключ сервера из **“Server Private Key”**;
- вместо **ens33** необходимо указать имя своей сетевой карты;
- параметр **PublicKey** необходимо изменить на свой публичный ключ клиента из **“Client Private Key”**;
- параметр **PresharedKey** необходимо изменить на свой сгенерированный PSK ключ из **“Client Pre-Shared Key”**.

Вносим необходимые измерения и сохраняем файл.

5. Автозагрузка и запуск VPN сервера WireGuard

Добавляем в автозагрузку, где **wg0** — имя файла конфигурации без указания расширения **.conf**

```
sysrc wireguard_enable="YES" && sysrc wireguard_interfaces="wg0"
```

VPN сервера WireGuard добавлен составной командой sysrc wireguard_enable="YES" && s

Или вручную открываем файл **/etc/rc.conf**

```
nano /etc/rc.conf
```

и добавляем строки:

```
wireguard_enable="YES"
wireguard_interfaces="wg0"
```

Запускаем VPN сервера WireGuard

```
wg-quick up wg0
```

VPN сервера WireGuard запущен командой “wg-quick up wg0” в операционной системе Fr

Для остановки используем команду

```
wg-quick down wg0
```

6. Создание файла конфигурации WireGuard клиента

Создаем файл конфигурации WireGuard клиента **wg0-client.conf** в каталоге **/usr/local/etc/wireguard/**

```
nano /usr/local/etc/wireguard/wg0-client.conf
```

Файл конфигурации клиента создается на сервере для удобства, чтобы была возможность в случае чего скопировать его в WireGuard клиент или передать через QR-код. Необязательно создавать данный файл на сервере, его можно сразу создать в WireGuard клиенте на устройстве пользователя.

Описание основных параметров конфигурационного файла WireGuard клиента

```
# Файл конфигурации VPN клиента WireGuard
#
# Раздел "Interface" относится к настройкам клиента
[Interface]
# Приватный ключ клиента из значения "Client Private Key"
PrivateKey = QKlR2932N1k5K4OvQ82DqQQnfJa/kpr3NIHtj+loTk4=
# IP адрес клиента и маска внутренней сети VPN
Address = 192.168.10.2/24
# Публичные DNS сервера или если на сервера VPN запущен DNS,
# то указываем IP адрес WireGuard сервера внутри VPN сети
DNS = 1.1.1.1, 1.0.0.1, 2606:4700:4700::1111, 2606:4700:4700::1001
#DNS = 192.168.10.1

# Раздел "Peer" относится к настройкам подключения к серверу
[Peer]
# Публичный ключ сервера из значения "Server Puplic Key"
PublicKey = Kyhl31NZO9hXhORc3hkpNMWFOF69ZCzlcQ4P70mh4xU=
# Ключ PSK из значения "Clinet Pre-Shared Key"
PresharedKey = G1WVZwGj+bDzggvqx9jBTL6NH4dhlW5rCW0PCFHY2oY=
# Публичный IP адрес сервера и порт, который был указан в файле wg0.conf
Endpoint = 80.95.110.25:51820
# До каких IP адресов пропускать трафик через VPN
# Весь трафик пускать через WireGuard
AllowedIPs = 0.0.0.0/0, ::/0
```

Используй всегда компактный файл, так как он необходим для генерации QR-кода и не везде поддерживается кириллица.

Компактный файл конфигурации клиента выглядит так:

```
[Interface]
PrivateKey = QKlR2932N1k5K4OvQ82DqQQnfJa/kpr3NIHtj+loTk4=
Address = 192.168.10.2/24
DNS = 1.1.1.1, 1.0.0.1, 2606:4700:4700::1111, 2606:4700:4700::1001

[Peer]
PublicKey = Kyhl31NZO9hXhORc3hkpNMWFOF69ZCzlcQ4P70mh4xU=
PresharedKey = G1WVZwGj+bDzggvqx9jBTL6NH4dhlW5rCW0PCFHY2oY=
Endpoint = 80.95.110.25:51820
AllowedIPs = 0.0.0.0/0, ::/0
```

Основные изменения которые необходимо внести в файл конфигурации:

- параметр **PrivateKey** необходимо изменить на свой приватный ключ клиента из **“Client Private Key”**;
- параметр **PublicKey** необходимо изменить на свой публичный ключ сервера из **“Server Private Key”**;
- параметр **PresharedKey** необходимо изменить на свой сгенерированный PSK ключ из **“Client Pre-Shared Key”**;
- в параметре **Endpoint** необходимо указать **IP адрес** и **порт** своего WireGuard клиента.

Вносим необходимые измерения и сохраняем файл.

7. Генерация QR-кода WireGuard туннеля

С телефона очень удобно получать файл конфигурации через сканирование QR-кода, но для этого файл **wg0-client.conf** необходимо преобразовать в QR-код. Утилита qrencode может создавать QR-коды и выводить их в консоль.

Устанавливаем утилиту **qrencode [libqrencode]**

```
pkg install libqrencode
```

Генерируем **QR-код** в терминале

```
qrencode -t ANSIUTF8 -r /usr/local/etc/wireguard/wg0-client.conf
```

где /**etc/wireguard/wg0-client.conf** — путь до файла конфигурации клиента.


Сгенерирован QR-код конфигурации WireGuard при помощи команды “qrencode -t ANSIU

8. Настройка файрвола и маршрутизации в FreeBSD

Чтобы была возможность раздавать интернет внутри VPN сети, необходимо настроить маршрутизацию пакетов через один из трех доступных файрволов (**IPFW**, **PF**, **IPF**) в

FreeBSD.

Если сервер является удаленным и без возможности получения доступа через дистанционную клавиатуру и монитор, то при настройке файрвола нужно быть очень внимательным. Так как если будет ошибка в настройке может получится так, что будет невозможно подключится через SSH.

*Я использую в качестве файрвола **IPFW**, поэтому покажу минимальные необходимые настройки на основе него, но также покажу еще пример настройки PF.*

8.1. Настройка IPFW для WireGuard

Включаем IP-форвардинг

```
sysctl net.inet.ip.forwarding=1
```

Открываем файл **/etc/rc.conf**

```
nano /etc/rc.conf
```

Добавляем строки

```
gateway_enable="YES" # Включить шлюз (сделает net.inet.ip.forwarding=1)
firewall_enable="YES" # Включить IPFW
firewall_nat_enable="YES" # Включить NAT
firewall_script="/etc/ipfw.rules" # Файл с правилами IPFW
```

Строку **gateway_enable="YES"** можно не добавлять, а вместо этого добавить строку **net.inet.ip.forwarding=1** в файл **/etc/sysctl.conf**

Открываем или создаем файл конфигурации **/etc/ipfw.rules**

```
nano /etc/ipfw.rules
```

Файл конфигурации IPFW + NAT для WireGuard VPN-сервера будет выглядеть примерно так:

```
#!/bin/sh
# Очищаем список правил IPFW
ipfw -q -f flush
# Команда добавления правила, -q — режим без вывода сообщений
```

```
cmd="ipfw -q add"
# Сетевые интерфейсы заносим в переменные
inet_if="em0" # сетевая карта интернета
wg_if="wg0" # интерфейс WireGuard (если wg0.conf, то указывать wg0)

# Разрешить локальный трафик
$cmd allow all from any to any via lo0

# Разрешить любые пакеты внутри сети WireGuard
$cmd allow ip from any to any via $wg_if

# Разрешить доступ к SSH серверу по 22 порту
$cmd allow tcp from any to me 22 in via $inet_if

# Разрешить доступ к VPN-серверу по протоколу UDP на 51820 порт
$cmd allow udp from any to me 51820 in via $inet_if

# Настройка маршрутизации для WireGuard
ipfw -q nat 1 config if $inet_if same_ports unreg_only deny_in reset
# Разрешить проходить любым пакетам через маршрутизацию
$cmd nat 1 all from any to any via $inet_if
```

Этих настроек будет достаточно для раздачи интернета внутрь VPN сети и будет возможность подключится к WireGuard и SSH из интернета. Весь исходящий трафик с сервера в интернет будет идти через правило маршрутизации.

Вносим изменения и сохраняем файл с правилами IPFW.

Будь внимателен. Если есть ошибка или неточность в правилах, можно потерять доступ по SSH.

Запускаем службу IPFW

```
service ipfw start
```

Можно переходить к проверке VPN-сервера WireGuard.

Более подробно о настройке IPFW читай: Настройка файрвола IPFW в FreeBSD и Настройка IPFW + NAT в FreeBSD

8.2. Настройка PF для WireGuard

Включаем IP-форвардинг

```
sysctl net.inet.ip.forwarding=1
```

Открываем файл **/etc/sysctl.conf**

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

добавляем строку **net.inet.ip.forwarding=1** для постоянного включения IP-форвардинга

```
net.inet.ip.forwarding=1
```

Сохраняем файл и закрываем.

Открываем файл **/etc/rc.conf**

```
nano /etc/rc.conf
```

Добавляем строки

```
pf_enable="YES" # Включить PF
pf_rules="/etc/pf.conf" # Файл с правилами PF
```

Открываем или создаем файл конфигурации **/etc/pf.conf**

```
nano /etc/pf.conf
```

Файл конфигурации PF + NAT для WireGuard VPN-сервера будет выглядеть примерно так:

```
# Определение интерфейсов и параметров
inet_if="em0" # сетевая карта интернета
wg_if="wg0" # интерфейс WireGuard (если wg0.conf, то указывать wg0)
wg_net = "192.168.10.0/24"  # Внутренняя сеть WireGuard (которая указана в wg0.conf)

# Пропускать локальный трафик
set skip on lo0
```

```
# Включение NAT для трафика из сети WireGuard
nat on $inet_if from $wg_net to any -> ($inet_if)

# Разрешить любые пакеты внутри сети WireGuard
pass on $wg_if all

# Разрешить доступ к SSH серверу по 22 порту
pass in on $inet_if proto tcp from any to ($inet_if) port 22

# Разрешить доступ к VPN-серверу по протоколу UDP на 51820 порт
pass in on $inet_if proto udp from any to ($inet_if) port 51820

# Разрешить весь исходящий трафик
pass out all
```

Этих настроек будет достаточно для раздачи интернета внутрь VPN сети и будет возможность подключится к WireGuard и SSH из интернета.

Вносим изменения и сохраняем файл с правилами PF.

Будь внимателен. Если есть ошибка или неточность в правилах, можно потерять доступ по SSH.

Запускаем службу PF

```
service pf start
```

Можно переходить к проверке VPN-сервера WireGuard.

9. Подключение к серверу WireGuard

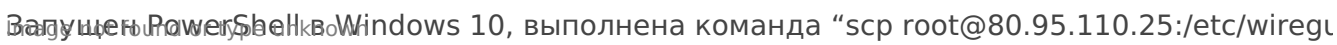
Теперь необходимо подключится к VPN сервера WireGuard и проверить его работоспособность с различных устройств. Если что-то не будет работать, необходимо внимательно проверить настройки и корректность внесения ключей.

9.1. Подключение из Windows к VPN серверу WireGuard

Запускаем **PowerShell** и копируем файл **wg0-client.conf** с сервера через утилиту **scp** в любое место на компьютере, например на рабочий стол

```
scp root@80.95.110.25:/etc/wireguard/wg0-client.conf 'C:\Users\User\Desktop\'
```

Вместо **root@80.95.110.25** указываем свои данные для доступа по SSH, вместо **User** указываем свое имя пользователя Windows.



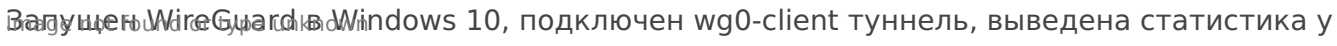
Запущен PowerShell в Windows 10, выполнена команда “scp root@80.95.110.25:/etc/wiregu

Скачиваем с официального сайта wireguard-installer.exe и запускаем установщик.
Произойдет установка WireGuard и он автоматически запустится.

Нажимаем на кнопку **Импорт туннелей из файла** и выбираем файл, скачанный ранее.

Запущен WireGuard в Windows 10, нажата кнопка “Импорт туннелей из файла”, открыто

Подключаемся к WireGuard серверу нажав кнопку **Подключить** и пробуем куда-нибудь зайти.



Запущен WireGuard в Windows 10, подключен wg0-client туннель, выведена статистика у

Как можно видеть, трафик принимается и передается, VPN WireGuard успешно работает.

Если интернет не работает, то проверяем что пакеты доходят до IP адреса сервера внутри VPN сети и до IP адреса в интернете

```
ping 192.168.10.1; ping 1.1.1.1
```

где **192.168.10.1** — IP адрес WireGuard сервера указанный в файле **wg0.conf** в параметре Address, а IP адрес **1.1.1.1** — публичный DNS сервер.

9.2. Подключение из Linux к VPN серверу WireGuard

Устанавливаем **WireGuard**

```
apt install wireguard
```

Копируем файл **wg0-client.conf** с сервера в каталог **/etc/wireguard/** через **scp**

```
scp root@80.95.110.25:/etc/wireguard/wg0-client.conf /etc/wireguard/
```

Подключаемся к WireGuard серверу

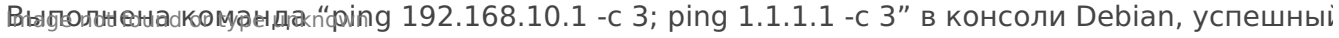
```
wg-quick up wg0-client
```

где **wg0-client** — имя файла конфигурации без указания расширения **.conf**

Проверяем что пакеты доходят до IP адреса сервера внутри VPN сети и до IP адреса в интернете

```
ping 192.168.10.1 -c 3; ping 1.1.1.1 -c 3
```

где **192.168.10.1** — IP адрес WireGuard сервера указанный в файле **wg0.conf** в параметре Address, а IP адрес **1.1.1.1** — публичный DNS сервер.


Выполнена команда “ping 192.168.10.1 -c 3; ping 1.1.1.1 -c 3” в консоли Debian, успешный

Как можно видеть, пинг до VPN сервера WireGuard и публичного DNS 1.1.1.1 успешен, можно пользоватся VPN WireGuard.

Чтобы выключить VPN, используем команду

```
wg-quick down wg0-client
```

Если необходимо, чтобы VPN подключался при запуске компьютера, то добавляем в автозагрузку автоматическое подключение к серверу WireGuard

```
systemctl enable wg-quick@wg0-client
```

Чтобы убрать из автозагрузки автоматическое подключение WireGuard, используем команду

```
systemctl disable wg-quick@wg0-client
```

9.3. Подключение с телефона Android к VPN серверу WireGuard

Устанавливаем приложение **WireGuard** из Google Play.

Запускаем приложение, нажимаем на + в нижней правой части экрана, выбираем из списка **Сканировать QR-код** и сканируем **QR-код** из консоли. Можно загрузить файл **wg0-client.conf** на телефон, выбрать из списка **Импорт из файла или архива** и указать данный файл.

Приложение WireGuard запущено на Android телефоне, нажата кнопка добавить туннели

Подключаемся к WireGuard серверу и пробуем куда-нибудь зайти.

Приложение WireGuard запущено на Android телефоне, выведена статистика успешного

Как можно видеть, в статистике трафик принимается и передается, VPN WireGuard успешно работает.

Revision #1
Created 4 July 2025 11:00:21 by buzz
Updated 4 July 2025 11:01:12 by buzz