{jcomments on}Связываем локальные сети при помощи OpenVPN

Имеем центральный офис и три филиала. Все используют разных интернет-провайдеров, и разные технологии.

Локальные сети:

Центральный офис: 10.10.10.0 255.255.255.0

Филиал под условным названием npn: 192.168.0.0 255.255.255.0

Филиал под условным названием mg: 192.168.10.0 255.255.255.0

Филиал под условным названием westfood: 192.168.2.0 255.255.255.0

Необходимо связать эти сети, для обмена данными 1С между филиалами и центральным офисом, ну и для удобства администрирования. Для этих целей будем использовать кросс-платформенную программу OpenVPN. С помощью нее создадим защищенную виртуальную частную сеть, обьединяющую локальные сети предприятия. Выглядеть это

будет так: из любой локальной сети, например с сети westfood (192.168.2.0 255.255.255.0) можно будет попасть на компьютер в любой другой сети, например npn(192.168.0.0 255.255.255.0), введя локальный ip-адрес, например 192.168.0.10 . Причем не важно что у них разные провайдеры. Программа использует только один порт, и не помеха если клиенты за NATом.

Ставим (и на сервере и на клиентах):

sudo su

apt-get install openvpn

Создание ключей (производится только на сервере. затем необходимые ключи копируются с сервера на клиенты).

Переходим в каталог со скриптами создания ключей шифрования:

cd /usr/share/doc/openvpn/examples/easy-rsa/2.0

Открываем файл, содержащий переменные для скриптов:

nano vars

изменяем следующие параметры под свою организацию:

export KEY_COUNTRY="RU"

export KEY_PROVINCE="PS"

export KEY_CITY="Pskov"

export KEY_ORG="MegaHolod"

export KEY_EMAIL="123@yandex.ru"

export KEY_DIR="/etc/openvpn/keys"

Последний - директория, куда будут сохраняться созданные ключи.

Заносим переменные из только что отредактированного файла в память

source ./vars

Перед созданием ключей запускаем скрипт:

./clean-all

Далее переходим непосредственно к генерированию ключей путем запуска соответствующих скриптов. Так как в файл с переменными мы уже занесли нужные значения, жмем просто Enter в ответ на вопросы скриптов, за исключением:

Sign the certificate? [y/n]:**y**

1 out of 1 certificate requests certified, commit? [y/n]**y**

Итак создаем CA ключ:

./build-ca

Создаем DH ключ (нужен только серверу):

./build-dh

Создаем private key для сервера (gate - имя сервера):

./build-key-server gate

Создаем ключи в PKCS #12 формате для машин-клиентов;

./build-key-pkcs12 mg

./build-key-pkcs12 npn

./build-key-pkcs12 westfood

Создаем TLS-ключ (Общий для сервера и клиента):

openvpn --genkey --secret /etc/openvpn/keys/ta.key

Из папки "/etc/openvpn/keys" нужно скопировать ta.key и *.p12 соответствующий клиенту на машины-клиенты.

Настраиваем сервер (создаем файл-конфиг и заполняем его):

touch /etc/openvpn/server.conf

nano /etc/openvpn/server.conf

port 17993 # порт, на котором будет слушать сервер

proto tcp # протокол (по умолчанию udp)

dev tun # тип устройства (tun или tap)

############################## KEYS ###

tls-server # явно указывает, что данный хост является tls-server

tls-auth keys/ta.key 0 # 0-сервер , 1- для конфига клиента

ca /etc/openvpn/keys/ca.crt # файл сертификата для CA

cert /etc/openvpn/keys/gate.crt # сертификат сервера

key /etc/openvpn/keys/gate.key # ключ сервера

dh /etc/openvpn/keys/dh1024.pem # файл с ключем Диффи-Хелмана

############################ END KEYS #####################################

 # автоматически присваивает адреса всем клиентам (DHCP) в указанном

 # диапазоне с маской сети. Данная опция заменяет ifconfig и может

 # работаеть только с TLS-клиентами в режиме TUN, соответственно

 # использование сертификатов обязательно.

server 10.8.0.0 255.255.255.0

ifconfig-pool-persist ipp.txt # Тут будут храниться ip адреса клиентов

push "route 10.10.10.0 255.255.255.0" # передача клиенту маршрута к сетке,

 # в которой сервер.

 # каждые 10 секунд посылать ping на удаленный хост, и, если за 60 секунд

 # не было получено ни одного пакета - то перезапускать туннель.

keepalive 10 60

 # параметр сжатия трафика, идущего через виртуальный туннель.

 # Может принимать значения yes, no, adaptive.

 # Последнее используется по умолчанию.

comp-lzo

 # Для улучшения безопасности рекомендовано запускать

 # все сервисы с минимальными правами. Openvpn будет работать от имени nobody.

user nobody

group nogroup

persist-key # указывает не перечитавать файлы ключей при перезапуске туннеля

persist-tun # данная опция оставляет без изменения устройства tun/tap

#при перезапуске OpenVPN.

 # сервер работает в режиме демона

daemon

############################## LOGS ##

status openvpn-status.log # указывает путь к статус-файлу,

 # в котором содержится информация о текущих соединениях и

 # информация о интерфейсах TUN/TAP

log-append openvpn.log # дописывать сообщения в лог-файл, а не перезаписывать.

verb 4 # уровень логирования

mute 20 # в лог будет записываться только по 20 сообщений из одной категории

############################ END LOGS ####################################

client-to-client # позволяет клиентам видеть друг друга (сети)

client-config-dir /etc/openvpn/ccd # папка содержащая маршруты к сетям

 # клиентов и посылаемые клиентам ip адреса клиента и сервера

ccd-exclusive # каждому клиенту свои настройки

management localhost 7505

tun-mtu 1500 # устанавливает максимальный размер MTU

tun-mtu-extra 32

mssfix 1450

 # маршруты к сетям клиентов

route 192.168.10.0 255.255.255.0 10.8.0.2

route 192.168.0.0 255.255.255.0 10.8.0.2

route 192.168.2.0 255.255.255.0 10.8.0.2

Настройки маршрутизации к клиентским сетям и выдача им ip.

Выбранные пары IP-адресов, во-первых, должны быть уникальными, во-вторых, должны входить в состав последовательных подсетей, ограниченных маской /30 (255.255.255.252), и, в-третьих, должны находиться в пределах пула IP-адресов, выделенного для виртуальной частной сети (определяется параметром server файла конфигурации сервера OpenVPN). С учетом перечисленных условий для клиентов и сервера подойдут пары IP-адресов со следующими парами последних октетов:

[1, 2] [5, 6] [9, 10] [13, 14] [17, 18] [21, 22] [25, 26] [29, 30] [33, 34] [37, 38]

[41, 42] [45, 46] [49, 50] [53, 54] [57, 58] [61, 62] [65, 66] [69, 70] [73, 74] [77, 78]

[81, 82] [85, 86] [89, 90] [93, 94] [97, 98] [101,102] [105,106] [109,110] [113,114] [117,118]

[121,122] [125,126] [129,130] [133,134] [137,138] [141,142] [145,146] [149,150] [153,154] [157,158]

[161,162] [165,166] [169,170] [173,174] [177,178] [181,182] [185,186] [189,190] [193,194] [197,198]

[201,202] [205,206] [209,210] [213,214] [217,218] [221,222] [225,226] [229,230] [233,234] [237,238]

[241,242] [245,246] [249,250] [253,254]

Создаем на сервере файлы для каждого клиента:

touch /etc/openvpn/ccd/npn

nano /etc/openvpn/ccd/npn

iroute 192.168.0.0 255.255.255.0

ifconfig-push 10.8.0.5 10.8.0.6 255.255.255.252

iroute - маршрут к сети клиента под названием npn

ifconfig-push <ip клиента> <ip сервера> <маска /30>

посылает клиенту ай-пи адрес клиента и сервера

для других сетей аналогично:

touch /etc/openvpn/ccd/mg

nano /etc/openvpn/ccd/mg

iroute 192.168.10.0 255.255.255.0

ifconfig-push 10.8.0.9 10.8.0.10 255.255.255.252

touch /etc/openvpn/ccd/westfood

nano /etc/openvpn/ccd/westfood

iroute 192.168.2.0 255.255.255.0

ifconfig-push 10.8.0.13 10.8.0.14 255.255.255.252

Правила для IPTABLES.

Для того что-бы это все работало, в фаерволе (iptables), если он используется, нужно разрешить трафик. Пример куска моего скрипта настройки, касающегося openvpn:

#!/bin/bash

###

 # Переменные

###

#указываем внешний ip сервера и внешн. сетевой интерфейс

INET_IP1=195.239.136.xxx

INET_IFACE1=eth2

 # указываем внутренний ip сервера и внутр. сетевой интерфейс

LAN_IP=10.10.10.4

LAN_IFACE=eth0

 # указываем сетевой интерфейс VPN, и сеть, ему принадлежащую

VPN_IFACE=tun0

VPN_RANGE=10.8.0.0/24

 # внутренняя сеть

LAN_RANGE=10.10.10.0/24

 # сетевой интерфейс петли и ip

LO_IFACE=lo

```
LO_IP=127.0.0.1

#######################################################

#OpenVPN

#######################################################

# разрешаем трафик между локальной сетью и VPN

# (необходимо для возможности доступа к серверу по внутреннему ip. с клиента)

$ip -A FORWARD -p all -i $LAN_IFACE -o $VPN_IFACE -j ACCEPT

$ip -A FORWARD -p all -o $LAN_IFACE -i $VPN_IFACE -j ACCEPT

# разрешаем входящий и исходящий трафик для vpn-интерфейса

# (необходимо для возможности установки vpn соединения)
```

$ip -A INPUT -p all -i $VPN_IFACE -j ACCEPT

$ip -A OUTPUT -p all -o $VPN_IFACE -j ACCEPT

 # разрешаем icmp пакеты через vpn

 # (необходимо для пинга)

$ip -A INPUT -p icmp -m icmp -i $VPN_IFACE --icmp-type echo-request -j ACCEPT

$ip -A OUTPUT -p icmp -m icmp -o $VPN_IFACE --icmp-type echo-request -j ACCEPT

$ip -A FORWARD -p icmp -m icmp -i $VPN_IFACE -o $LAN_IFACE --icmp-type echo-request -j ACCEPT

$ip -A FORWARD -p icmp -m icmp -o $VPN_IFACE -i $LAN_IFACE --icmp-type echo-request -j ACCEPT

$ip -A INPUT -p icmp -m icmp -i $VPN_IFACE --icmp-type echo-reply -j ACCEPT

$ip -A OUTPUT -p icmp -m icmp -o $VPN_IFACE --icmp-type echo-reply -j ACCEPT

$ip -A FORWARD -p icmp -m icmp -i $VPN_IFACE -o $LAN_IFACE --icmp-type echo-reply -j ACCEPT

$ip -A FORWARD -p icmp -m icmp -o $VPN_IFACE -i $LAN_IFACE --icmp-type echo-reply -j ACCEPT

Теперь на машинах-клиентах создаем конфиг:

touch /etc/openvpn/client.conf

nano /etc/openvpn/client.conf

client

dev tun # тип устройства tun или tap

#dev-node OpenVPN # раскоментировать если клиент под виндовс

 # для виндовс нужно указать название

 # создаваемого сетевого адаптера

proto tcp # протокол. по умолчанию udp

remote 195.239.136.xxx 17993 # ip и порт сервера

remote 93.153.252.xxx 17993

resolv-retry infinite # для dyndns

persist-key # указывает не перечитавать файлы ключей при перезапуске туннеля

persist-tun # данная опция оставляет без изменения устройства tun/tap

################KEY######################

tls-client

 # пути к файлам ключей, которые мы скопировали с сервера

pkcs12 /etc/openvpn/keys/westfood.p12

tls-auth /etc/openvpn/keys/ta.key 1

##############END_KEY#####################

 # параметр сжатия трафика, идущего через виртуальный туннель.

 # Может принимать значения yes, no, adaptive.

 # Последнее используется по умолчанию.

comp-lzo

verb 4 # уровень логирования

tun-mtu 1500 # устанавливает максимальный размер MTU

tun-mtu-extra 32

mssfix 1450

route-delay 5 # посылать маршруты через 5 сек. после установки vpn-канала

management localhost 7505

 # show-net-up # раскоментировать, если клиент под виндовс

 # если в конфиг OpenVPN вставить show-net-up, то OpenVPN запросит

 # windows через API всю таблицу маршрутизации и выведет её в лог

#ip-win32 manual # раскоментировать, если клиент под виндовс

Запускаем на сервере, а затем и на клиентах демон openvpn:

/etc/init.d/openvpn start

Выходим из под рута:

exit