VPN через Yggdrasil

Предисловие

У меня есть свой vps у зарубежного хостера. На нём успешно несколько лет работал WireGuard (далее WG). Но пришло зло и начало блокировать мой уютный `vpn`. Я начал думать и гадать как же дальше буду...

Первое что пришло в голову это обфускация трафика, очевидно что WG блокируют по сигнатуре в трафике.

Да, это как раз произошло в тот момент когда я начал щупать за "вымя" великолепный Yggdrasil.

Поэтому ответ напрашивался сам собой. У меня же и так есть "обфускатор" в виде этого Yggdrasil. Почему бы не отказаться от WG и просто гнать локальный трафик через Yggdrasil на vps а дальше "маскарадингом" выпускать наружу.

Основная проблема состояла в том, что Yggdrasil внутри своей сети использует исключительно `ipv6`, а трафик с локальной машины у меня (как в прочем и у всех) `ipv4`. И нужно было как-то его, этот трафик пропихнуть через Yggdrasil. Немного "погуглив", я нашел довольно простой способ сделать это без установки и конфигурации какого-либо дополнительного софта.

Да, большой плюс Yggdrasil состоит в том, что его сеть является одной большой локальной сетью. По сути это один большой Broadcast Domain. Все в сети могут напрямую по `ipv6` видеть всех остальных.

Так вот оказалось, что все для организации проброса уже есть под рукой практически в любом дистрибутиве Linux. Это утилита ip. Да-да, именно та утилита, которой мы смотрим наши ip адреса на интерфейсах.

С помощью этой утилиты мы можем создать ip туннель в режиме `ipip6`, который будет инкапсулировать наши пакеты `ipv4` в пакеты `ipv6` и отправлять их через туннель, который будет работать поверх сети Yggdrasil. Пакеты будут приходить в конечную точку тоннеля, на наш vps. Выходя из тоннеля пакеты будут декапсулироваться (хз есть ли такое слово) и передаваться на `ipv4` интерфейс вашей vps для дальнейшего путешествия по сети интернет.

Локальная машина

Создание туннеля

Давайте произведём настройки на стороне клиента. У вас уже должна быть установлена и настроена сеть Yggdrasil.

Выполним следующую команду на вашем локальном компьютере:

```
> ip -br a show tun0
tun0             UNKNOWN        200:d4a5:ec3f:5914:274c:8f70::660/7
```

Увидим примерно следующий вывод. `tun0` это виртуальный интерфейс сети Yggdrasil.

Тут нас интересует ipv6 адрес выданный нам этой сетью:

`200:d4a5:ec3f:5914:274c:8f70::660` - Это будет у нас Local IP

Выполним туже самую команду на нашем vps:

```
> ip -br a show tun0
tun0             UNKNOWN        200:d4a5:ec3f:5914:274c:8f70::1/7
```

И запишем ip адрес:

`200:d4a5:ec3f:5914:274c:8f70::1` - Это будет у нас Remote IP

Создадим туннель на локальной машине:

```
sudo ip link add name ipip6 type ip6tnl local 200:d4a5:ec3f:5914:274c:8f70::660
remote 200:d4a5:ec3f:5914:274c:8f70::1 mode ipip6
```

После создания туннеля у нас появятся два новых интерфейса:

```
> ip -br a
ip6tnl0@NONE     DOWN
tun0             UNKNOWN        200:d4a5:ec3f:5914:274c:8f70::660/7
ipip6@NONE       DOWN
```

Работать мы будем с интерфейсом `ipip6`

Включим этот интерфейс:

```
sudo ip link set ipip6 up
```

Далее вам нужно будет определиться с адресом подсети `ipv4`, которая будет работать внутри туннеля.
Это может быть например `192.168.44.0/30`. Из этой подсети с такой маской мы можем взять только два адреса (а больше в нашем случае и не нужно). Первый для конца туннеля на локальной машине, второй для конца туннеля на удаленном vps.

Пускай это будут:

```
192.168.44.1 #Для удаленного vps
192.168.44.2 #Для локальной машины
```

Назначим адрес `ipv4` для нашего туннеля:

```
sudo ip address add 192.168.44.2/30 dev ipip6
```

Маршрутизация

Теперь нам нужно отправить наш ipv4 трафик на vps. На локальной машине нам нужно будет создать пару маршрутов.

Подставьте вместо `XX.XX.XX.XX/32` ip адрес вашего пира, а вместо `YY.YY.YY.YY` ip адрес вашего домашнего роутера, `wifi0` - замените на интерфейс на котором у вас ваша домашняя сеть.

```
sudo ip route add XX.XX.XX.XX/32 via YY.YY.YY.YY dev wifi0
```

Тут нужно пояснить. У вас в Yggdrasil прописан как минимум один `peer` через которого вы выходите в сеть.
Так вот нам нужно что-бы соединение с этим пиром шло обязательно через "клирнет", не через сеть Yggdrasil.
Потому-что мы не можем соединиться с сетью Yggdrasil через неё же саму, получается зацикливание. Из-за этого ничего работать не будет.

Для этого нам и нужен маршрут, который будет всегда "роутить" соединение до пира через обычный интернет. Как раз это правило маршрутизации нам в этом поможет.

Далее прописываем следующий маршрут. Этот маршрут будет отправлять весь остальной трафик через наш туннель на vps.

```
sudo ip route add 0.0.0.0/0 via 192.168.44.2 dev ipip6
```

Удаленный VPS

Создание туннеля

Переходим к нашей vps. Настроим туннель со стороны vps. Настройка будет точно такая же, разница только в `ip` адресах.
При создании туннеля нужно поменять `ipv6` адреса местами `Local IP <---> Remote IP`

```
sudo ip link add name ipip6 type ip6tnl local 200:d4a5:ec3f:5914:274c:8f70::1
remote 200:d4a5:ec3f:5914:274c:8f70::660 mode ipip6
```

Со стороны vps так же назначаем `ipv4` на интерфейс:

```
sudo ip address add 192.168.44.1/30 dev ipip6
```

Настройка iptables

Теперь мы напишем правила для `iptables`, которые проведут наш трафик из туннеля в интернет.

Вместо `eth0` подставьте интерфейс с которого у вас выход в интернет.

```
# Эти два правила разрешат пересылку пакетов между интерфейсами
sudo iptables -A FORWARD -i eth0 -o ipip6 -j ACCEPT
sudo iptables -A FORWARD -i ipip6 -o eth0 -j ACCEPT

# Это правило подменит локальный ip адрес (192.168.44.2) при выходе из интерфейса
eth0 на внешний ip
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Проверка работоспособности

Всё, туннель у нас должен работать. Для проверки работоспособности "попингуйте" друг друга через туннельные ip:

```
ping 192.168.44.1  #На локальной машине
ping 192.168.44.2  #На удаленной машине
```

Если всё "пингуется" в обе стороны, то туннель успешно работает. Далее проверьте, что интернет работает именно через туннель.

```
# Здесь мы видим что трафик уходит через ip нашего туннеля
> traceroute -n 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.100.44.1  93.791 ms  98.394 ms  98.443 ms
 2  49.11.8.1  98.441 ms  98.434 ms  98.419 ms
 ...
```

Заключение

После этого у нас на локальной машине должен работать интернет через туннель, который проходит через сеть Yggdrasil.

Стоит напомнить что все эти настройки сбросятся после перезагрузки. Так что вам необходимо их прописать в конфигурационные файлы ваших дистрибутивов Linux.

Для интерфейсов это можно сделать через `Systemd-Networkd` с помощью файлов `netdev`. А с помощью файлов `network` можно назначить `ip` адреса на интерфейсы.

Для `iptables` правила можно прописать в файле `/etc/iptables/iptables.conf`

shedar@riseup.net