Инструкция: как сделать удаленный доступ web интерфейсу роутера (Luci) на OpenWrt (даже если он находится за NAT и у него «серый» IP адрес), используя FRP (fast reverse proxy). Как альтернатива инструментам удаленного доступа: Zerotier или Tailscale.

Необходимо иметь VPS с белым IPv4 (пусть в примере будет 111.111.111.111)

VPS будет использоваться как средство обхода блокировок и для доступа в вебморде роутера (из любого места).

Подключаемся к VPS (Debian 12) по ssh, используя Putty (или аналог)

Обновляем информацию о репозиториях и обновляем установленные пакеты:

apt-get update && apt-get upgrade -y

apt install net-tools

1. **Часть: Устанавливаем на VPS Xray (от root, с геоданными)**

bash -c "$(curl -L https://github.com/XTLS/Xray-install/raw/main/install-release.sh)" @ install -u root

Генерируем с помощью Xray необходимые для работы VLESS-Reality аутентификационные параметры:

xray uuid - команда сгенерирует UUID, это что-то типа логина пользователя

xray x25519 – команда сгенерирует приватный и публичный ключ сервера

Запишем их (для xray):

uuid: **ff19ee23-80ac-4997-9f75-a936508653b2**

Private key: **ZV-59M9rxU9JtHeQfK-hQRNxoP2-AOnIBc1HozAJ8xQ**

Public key: **VeqIvPfdg2hHvrsBKh6Ug37rf7bUvMsIE1cs-bMjbye**

Приватный ключ используется в серверном конфиге Xray (на VPS) , а публичный в клиентском (на телефоне); будьте внимательны

С помощью WinSCP заходим на VPS, идем в папку /usr/local/etc/xray/config.json и вставляем:

==========

```
{
  "log": {
    "loglevel": "info"
  },
  "inbounds": [
    {
      "listen": "IP_адрес_вашего_сервера",
      "port": 443,
      "protocol": "vless",
      "tag": "reality-in",
      "settings": {
        "clients": [
          {
            "id": "ваш_UUID",
            "email": "user1",
            "flow": "xtls-rprx-vision"
          }
        ],
```

```
        "decryption": "none"
      },
      "streamSettings": {
        "network": "tcp",
        "security": "reality",
        "realitySettings": {
          "show": false,
          "dest": "ваш_маскировочный_домен:443",
          "xver": 0,
          "serverNames": [
            "ваш_маскировочный_домен"
          ],
          "privateKey": "ваш_ПРИВАТНЫЙ_ключ",
          "minClientVer": "",
          "maxClientVer": "",
          "maxTimeDiff": 0,
          "shortIds": [""]
        }
      },
      "sniffing": {
        "enabled": true,
        "destOverride": [
          "http",
          "tls",
          "quic"
        ]
      }
    }
  ],
  "outbounds": [
    {
      "protocol": "freedom",
      "tag": "direct"
    },
    {
      "protocol": "blackhole",
      "tag": "block"
    }
  ],
  "routing": {
    "rules": [
      {
        "type": "field",
        "protocol": "bittorrent",
        "outboundTag": "block"
      }
    ],
    "domainStrategy": "IPIfNonMatch"
  }
}
```

==========

Маскировочный домен должен соответствовать критериям (**важно**): зарубежный незаблокированный с обоих сторон веб-сайт, поддерживающий TLS 1.3 и H2; адрес, без переадресации куда-либо еще (домен может быть перенаправлен на www). Пусть маскировочным будет сайт: **openstreetmaps.org**

Тогда json будет выглядеть так (внимательно с лишними пробелами при копировании в кавычках, запятыми, скобками и т.п; синтаксис можно проверить используя https://codebeautify.org/jsonvalidator):

==========

```
{
  "log": {
    "loglevel": "info"
  },
  "inbounds": [
    {
      "listen": "111.111.111.111",
      "port": 443,
      "protocol": "vless",
      "tag": "reality-in",
      "settings": {
        "clients": [
          {
            "id": "ff19ee23-80ac-4997-9f75-a936508653b2",
            "email": "user1",
            "flow": "xtls-rprx-vision"
          }
        ],
        "decryption": "none"
      },
      "streamSettings": {
        "network": "tcp",
        "security": "reality",
        "realitySettings": {
          "show": false,
          "dest": "openstreetmaps.org:443",
          "xver": 0,
          "serverNames": [
            "openstreetmaps.org"
          ],
          "privateKey": "ZV-59M9rxU9JtHeQfK-hQRNxoP2-AOnIBc1HozAJ8xQ",
          "minClientVer": "",
          "maxClientVer": "",
          "maxTimeDiff": 0,
          "shortIds": [""]
        }
      },
      "sniffing": {
        "enabled": true,
        "destOverride": [
          "http",
          "tls",
          "quic"
        ]
      }
    }
  ],
  "outbounds": [
    {
      "protocol": "freedom",
      "tag": "direct"
    },
    {
      "protocol": "blackhole",
      "tag": "block"
    }
  ],
  "routing": {
    "rules": [
      {
        "type": "field",
        "protocol": "bittorrent",
        "outboundTag": "block"
      }
    ],
```

```
    "domainStrategy": "IPIfNonMatch"
  }
}
================================================================================
```

Перезапустите XRay командой **systemctl restart xray**. Сразу после этого можно проверить что все нормально командой **systemctl status xray** (должно быть написано active (running).

Сделаем клиентский конфиг (для телефона). Шаблон такой:

vless://ваш_UUID@IP_адрес_вашего_сервера:443/?encryption=none&type=tcp&sni=домен_сайта&fp=chrome&security=reality&alpn=h2&flow=xtls-rprx-vision&pbk=**ваш_публичный_ключ**&packetEncoding=xudp

В примере получится так:

vless://**ff19ee23-80ac-4997-9f75-a936508653b2@111.111.111.111**:443/?encryption=none&type=tcp&sni=**openstreetmaps.org**&fp=chrome&security=reality&alpn=h2&flow=xtls-rprx-vision&pbk=**VeqIvPfdg2hHvrsBKh6Ug37rf7bUvMsIE1cs-bMjbye**&packetEncoding=xudp

Устанавливам на телефон NekoBox и вставляем наш конфиг. Запускаем наш прокси и тестируем его (ходим по заблокированному интернету, запустим speedtest и т.п; чтобы убедиться что xray работает хорошо)

2. **<u>Часть. Устанавливаем FRP</u>**

<u>Серверная часть</u>

Перезагружаем VPS и запускаем команду (не зависит от архитектуры процессора на VPS; можно и на Xeon Intel):

wget https://github.com/fatedier/frp/releases/download/v0.61.0/frp_0.61.0_linux_amd64.tar.gz

распаковываем:

tar -xvzf frp_0.61.0_linux_amd64.tar.gz

Появится папка с 4 файлами (если под root делали)

/root/frp_0.61.0_linux_amd64

Создадим для запуска FRP отдельный каталог:

mkdir -p /opt/frps

Зайдем в папку /root/frp_0.61.0_linux_amd64

cd /root/frp_0.61.0_linux_amd64

Выполним (находясь в папке /root/frp_0.61.0_linux_amd64) 2 команды:

mv frps /opt/frps/

mv frps.toml /opt/frps

Теперь создаём сервис:

nano /etc/systemd/system/frps.service

Вставляем в него следующее содержимое (что между разделителями):

==========

[Unit]

Description=FRP Server

After=network.target

[Service]

ExecStart=/opt/frps/frps -c /opt/frps/frps.toml

User=nobody

Group=nogroup

Restart=always

[Install]

WantedBy=multi-user.target

==========

Сохраняем Ctrl+O и закрываем Ctrl+X

Запускаем FRP

systemctl daemon-reload

systemctl enable frps

Теперь правим конфигурацию

nano /opt/frps/frps.toml

Вставляем содержимое (свои). Т.е. выбираем порт, на котором будет слушать frps и некое кодовое слово / ключ или auth.token генерируем, например, с помощью online генератора https://www.uuidgenerator.net/ или же можно командой на VPS xray uuid (тк xray у нас установлен и встроенную команду xray uuid можно использовать и для frp)

==========

bindPort = 63334

proxyBindAddr = "127.0.0.1"

auth.token = "658a2c30-9835-45fc-b18a-0824aa1914d8"

transport.tls.force = true

==========

Последняя строка **transport.tls.force = true** означает, что без tls соединение не будет установлено.

Полные настройки (если интересно) можно изучить на странице проекта frp https://github.com/fatedier/frp/blob/dev/conf/frps_full_example.toml

Сохраняем Ctrl+O и закрываем Ctrl+X

Перезапускаем FRPS (S на конце занчит server; C – client)

systemctl restart frps

Дальше можно заглянуть и убедиться что frps слушает порт

netstat -tanp | grep frps

И посмотреть в журнале лог запуска

journalctl -u frps

Если хочется дополнительной безопасности (типа в свои сети я хожу только из своих сетей; с левых посторонних ip на сервере файрвол ufw не пустит, то нужно на VPS открыть доступ для своих статических IP или ip других своих VPS, или хотябы для CIDR своих провайдеров мобильного и проводного интернета. Чтобы посторонние на сервер не ломились. Но это на ваше усмотрение. В этом примере я сильно заморачиваться не буду:

ufw allow in to any port 63334 proto tcp from XX.XX.XX.XX/XX comment 'frps'

с сервером почти все (в заключительной части еще чуть доделаем для большего удобства)

Клиентская часть FRPC OpenWRT роутер:

Зайти в Luci – System – Software (нажать update lists, затем в поле filter написать: luci-app-frpc и установить вместе с зависимостями)

Подключаемся к роутеру Openwrt по ssh, используя Putty (или аналог)

Редактируем файл (вставляем в него содержимое между разделителями)

nano /etc/config/frpc

==========

config init

    option stdout '1'

    option stderr '1'

    option user 'root'

    option group 'root'

    option respawn '1'

config conf 'common'

    option server_addr '**111.111.111.111**' # server ip VPS

    option server_port '**63334**' # frps port

    option log_level 'trace'

    option token **'658a2c30-9835-45fc-b18a-0824aa1914d8'** # тот же токен

config conf 'ssh'

option type 'tcp'

option local_ip **'192.168.1.1'** # внутр адрес роутера

option local_port '22'

option remote_port '10022' # порт который будет на сервере под ssh

option name 'ssh'

config conf 'web'

option type 'tcp'

option local_ip **'192.168.1.1'**

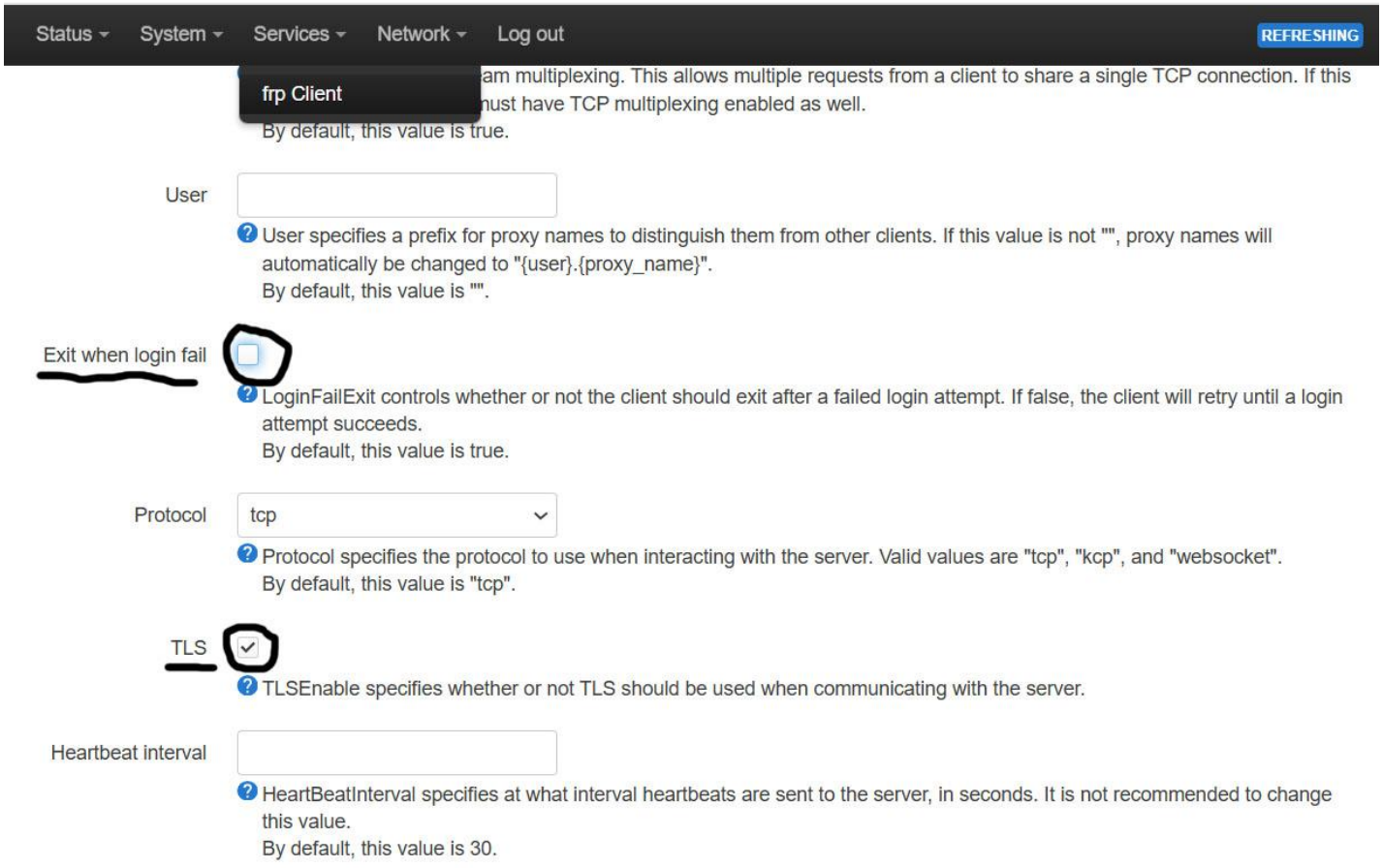
option local_port '80'

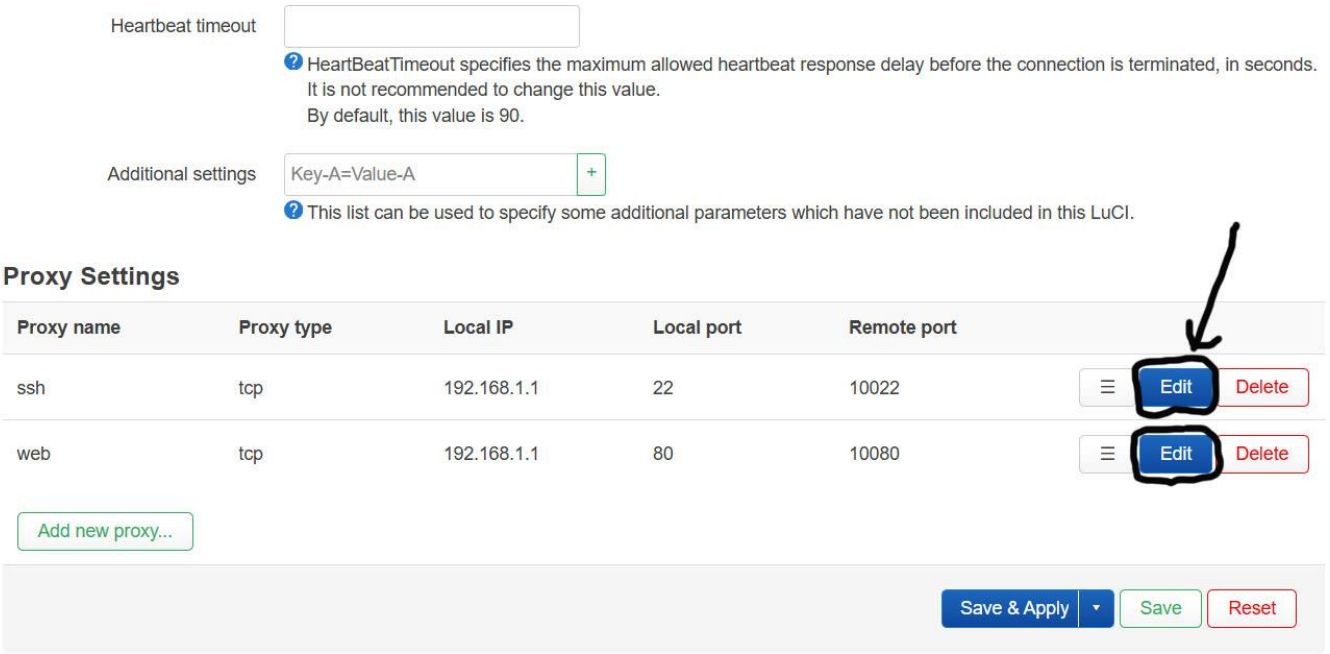
option remote_port '10080'

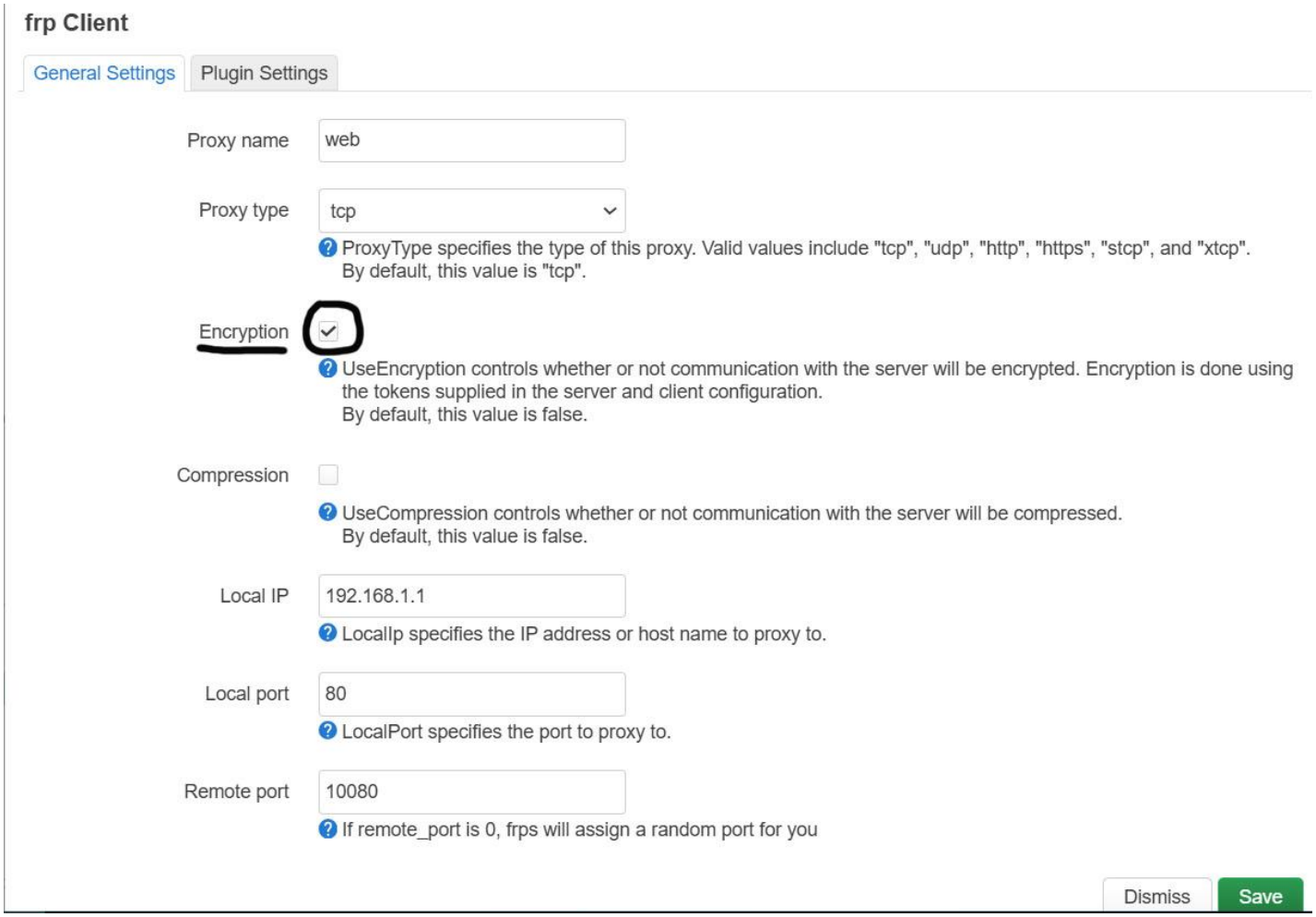
option name 'web'

==========

Заходим в Luci – Services – frp Client и ставим галочки как на скриншотах ниже







В обоих разделах (ssh и web) ставим галочку Encryption . После нажимаем Save and Apply

Перезагружаем frpc на роутере

service frpc enable && service frpc restart

После этого (примерно через минуту) на роутере

logread -e frpc

И на сервере

journalctl -u frps

Должны появится записи об успешном пробросе

Команда: **netstat -tanp | grep frps**

на сервере покажет, что слушаются порты 10022, 10080

В ssh роутера теперь попасть так: зайти с телефона по ssh на vps и там

ssh -p 10022 root@127.0.0.1

На веб интерфейс роутера Luci можно попадать красиво:

Будем использовать какой-нибудь сайт, который не требуется никогда. Например: neverssl.com

В правилах роутинга nekobox, foxray или чем пользуемся на телефоне сделаем, чтобы этот адрес шел в туннель (если точечная маршрутеризация; если же все идет в тунель, тогда не заморачиваемся)

в xray на сервере добавляем правило:

домен neverssl.com порт 80

переадресовать на 127.0.0.1:10080

Для Xray это будет выглядеть так:

```
================================================================================
"outbounds": [
...
    {
      "tag": "to_10080",
      "protocol": "freedom",
      "settings": {"redirect": "127.0.0.1:10080"}
    }
  ],
...

  "routing": {
    "rules": [
      {
        "domain": ["neverssl.com"],
        "port": 80,
        "outboundTag": "to_10080"
      }
...
================================================================================
```

С помощью WinSCP заходим на VPS, идем в папку /usr/local/etc/xray/config.json и вставляем эти дополнительные разделы. Получаем итоговый json (внимательно с лишними пробелами при копировании в кавычках, запятыми, скобками и т.п; синтаксис можно проверить используя https://codebeautify.org/jsonvalidator):

==========

```
{
  "log": {
    "loglevel": "Error"
  },
  "inbounds": [
    {
      "listen": "111.111.111.111",
      "port": 443,
      "protocol": "vless",
      "tag": "reality-in",
      "settings": {
        "clients": [
          {
            "id": "ff19ee23-80ac-4997-9f75-a936508653b2",
            "email": "user1",
            "flow": "xtls-rprx-vision"
          }
        ],
        "decryption": "none"
      },
      "streamSettings": {
        "network": "tcp",
        "security": "reality",
        "realitySettings": {
          "show": false,
          "dest": "openstreetmaps.org:443",
          "xver": 0,
          "serverNames": [
            "openstreetmaps.org"
          ],
          "privateKey": " ZV-59M9rxU9JtHeQfK-hQRNxoP2-AOnIBc1HozAJ8xQ",
          "minClientVer": "",
          "maxClientVer": "",
          "maxTimeDiff": 0,
          "shortIds": [""]
       }
      },
      "sniffing": {
        "enabled": true,
        "destOverride": [
          "http",
          "tls",
          "quic"
        ]
      }
```

```
    }
  ],
  "outbounds": [
    {
          "protocol": "freedom",
      "tag": "direct"
    },
    {
      "protocol": "blackhole",
      "tag": "block"
    },
    {
      "tag": "to_10080",
      "protocol": "freedom",
      "settings": {"redirect": "127.0.0.1:10080"}
    }
    ],
  "routing": {
    "rules": [
      {
        "domain": ["neverssl.com"],
        "port": 80,
        "outboundTag": "to_10080"
      },
      {
        "type": "field",
        "protocol": "bittorrent",
        "outboundTag": "block"
      }
    ],
    "domainStrategy": "IPIfNonMatch"
  }
}
```

==========

Теперь подключившись с телефона к VPS через NekoBox и введя в адресную строку браузера:

https://neverssl.com мы будем попадать в luci на web морду роутера OpenWRT.

Все!!!

P.S.

Если роутеров к которым нужно иметь доступ несколько? Конфиг сервера остается тот же

На втором клиенте указывать другие remote_port и option name

Т.е. например 11022, 11080 и имена ssh2, web2

Если вместо XRAY на VPS стоит 3X-UI ?

Xray есть в составе 3x-ui. Только как он называется xray-linux-что-то-там; можно найти так:
ps -e | grep xray

3x-ui просто прокладка графическая. В ней можно также добавить аутбонды и правила