Jacks Blog

I will have a life no remorse, in the future, we must accomplish something.

OpenWRT OpenVPN配置远程访问所有家里局域网服务

2018-12-30 | 2018-12-30 | 网络 | 6642 | 1,684

> 这段时间家里自建了NAS，做了很多服务，不过这类服务都是基于局域网的，如果只是通过在电信猫与路由器上进行端口转发，这样回到家使用内网IP内网的端口策略，出门后又要改用公网IP，公网端口策略十分不方便，于是想到了公司目前很多AWS服务都是基于OpenVPN对其局域网内的各类服务访问，于是就打算在OpenWRT上搭建一个OpenVPN服务，这样出门只需要通过Tunnelblick连接上路由器的VPN，将所有 `192.168.99.0/24` 的流量路由到OpenVPN上，就可以保证出门在外也和在家时候一样的访问所有家里局域网的服务，不用走两套策略了。

具体OpenVPN的各类配置特征可以直接参看OpenWrt的这个帖子，我们今天的整个流程也是主要参考该教程进行实践的，关于OpenWRT路由搭建相关的博客中有很多文章了，感兴趣的可以搜索查看。

I. 准备工作

先SSH登录到路由器OpenWRT上。安装必要的软件:

```
opkg update
opkg install openvpn-openssl luci-app-openvpn
```

II. 创建证书

主要是创建用于安全通信的证书，下面的步骤是连续的，一步一步的复制粘贴执行下面的指令便可以完成:

第一步. PKI目录

```
PKI_DIR="/etc/openvpn/ssl"
mkdir -p ${PKI_DIR}
chmod -R 0600 ${PKI_DIR}
cd ${PKI_DIR}
touch index.txt; echo 1000 > serial
mkdir newcerts
```

第二步. openssl配置文件

拷贝 `/etc/ssl/openssl.cnf` 作为基准:

```
cp /etc/ssl/openssl.cnf ${PKI_DIR}
```

修改必要的内容为目标配置:

```
PKI_CNF=${PKI_DIR}/openssl.cnf

sed -i '/^dir/   s:=.*:= /etc/openvpn/ssl:'                    ${PKI_CNF}
sed -i '/.*Name/ s:= match:= optional:'                   ${PKI_CNF}

sed -i '/organizationName_default/    s:= .*:= WWW Ltd.:'  ${PKI_CNF}
sed -i '/stateOrProvinceName_default/ s:= .*:= London:'    ${PKI_CNF}
sed -i '/countryName_default/         s:= .*:= GB:'        ${PKI_CNF}

sed -i '/default_days/   s:=.*:= 3650:'                   ${PKI_CNF} ## default usu.: -days 3
sed -i '/default_bits/   s:=.*:= 4096:'                   ${PKI_CNF} ## default usu.: -newkey
```

添加必要的内容:

```
cat >> ${PKI_CNF} <<"EOF"
################################################################################
### Check via: openssl x509 -text -noout -in *.crt | grep 509 -A 1
[ my-server ]
#  X509v3 Key Usage:          Digital Signature, Key Encipherment
#  X509v3 Extended Key Usage: TLS Web Server Authentication
  keyUsage = digitalSignature, keyEncipherment
  extendedKeyUsage = serverAuth

[ my-client ]
#  X509v3 Key Usage:          Digital Signature
#  X509v3 Extended Key Usage: TLS Web Client Authentication
  keyUsage = digitalSignature
  extendedKeyUsage = clientAuth

EOF
```

第三步. 创建服务端与客户端的文件

服务端文件生成:

```
openssl req -batch -nodes -new -keyout "ca.key" -out "ca.crt" -x509 -config ${PKI_CNF}  ## x509
openssl req -batch -nodes -new -keyout "my-server.key" -out "my-server.csr" -subj "/CN=my-serve
openssl ca  -batch -keyfile "ca.key" -cert "ca.crt" -in "my-server.csr" -out "my-server.crt" -c
```

客户端文件生成:

```
openssl req -batch -nodes -new -keyout "my-client.key" -out "my-client.csr" -subj "/CN=my-clien
openssl ca  -batch -keyfile "ca.key" -cert "ca.crt" -in "my-client.csr" -out "my-client.crt" -c
```

权限配置:

```
chmod 0600 "ca.key"
chmod 0600 "my-server.key"
chmod 0600 "my-client.key"
```

第四步. Diffie–Hellman生成

```
openssl dhparam -out dh2048.pem 2048
```

III. OpenVPN相关网络配置

1. 创建VPN接口(命名为vpn0)

```
uci set network.vpn0=interface
uci set network.vpn0.ifname=tap0
uci set network.vpn0.proto=none
uci set network.vpn0.auto=1
```

2. 添加接口到LAN桥中

```
uci set network.lan.ifname="$(uci get network.lan.ifname) tap0"
```

3. 允许客户端的进口的流量输入

这里我们都是使用 1194 这个openVPN的默认端口:

```
uci set firewall.Allow_OpenVPN_Inbound=rule
uci set firewall.Allow_OpenVPN_Inbound.target=ACCEPT
uci set firewall.Allow_OpenVPN_Inbound.src=*
uci set firewall.Allow_OpenVPN_Inbound.proto=udp
uci set firewall.Allow_OpenVPN_Inbound.dest_port=1194
```

4. 生效配置

```
uci commit network
/etc/init.d/network reload
```

```
uci commit firewall
/etc/init.d/firewall reload
```

IV. OpenVPN配置

将刚刚我们生成的一系列证书进行拷贝到OpenVPN配置目录中:

```
cp /etc/openvpn/ssl/ca.crt /etc/openvpn/ssl/my-server.* /etc/openvpn/ssl/dh2048.pem /etc/openvp
```

清空原本的配置并进行配置:

```
echo > /etc/config/openvpn
uci set openvpn.myvpn=openvpn
uci set openvpn.myvpn.enabled=1
uci set openvpn.myvpn.verb=3
uci set openvpn.myvpn.proto=udp
uci set openvpn.myvpn.port=1194
uci set openvpn.myvpn.dev=tap
uci set openvpn.myvpn.mode=server
uci set openvpn.myvpn.tls_server=1
uci add_list openvpn.myvpn.push='route-gateway dhcp'
uci set openvpn.myvpn.keepalive='10 120'
uci set openvpn.myvpn.ca=/etc/openvpn/ca.crt
uci set openvpn.myvpn.cert=/etc/openvpn/my-server.crt
uci set openvpn.myvpn.key=/etc/openvpn/my-server.key
uci set openvpn.myvpn.dh=/etc/openvpn/dh2048.pem
uci commit openvpn
```

配置开机启动并且启动服务

```
/etc/init.d/openvpn enable
/etc/init.d/openvpn start
```

此时我们可以直接通过LUCI中直接看到启动的服务:

V. 客户端配置

这边大家可以搜索下客户端可以使用 ovpn (openVPN)的客户端，这里由于我是Mac系统，我使用的是Tunnelblick，不过配置文件基本上都是通用的。

下面我们假设我们最终将 ovpn 文件放在 ~/openvpn 中(你可以放在任何你想要的目录)。

1. 拷贝客户端证书

我们将在OpenWRT上刚刚生成的 `/etc/openvpn/ssl/ca.crt` 、 `/etc/openvpn/ssl/my-client.key` 、 `/etc/openvpn/ssl/my-client.csr` 都拷贝到 `~/openvpn` 。

2. 拿到你的公网IP

可以通过cip.cc拿到你的公网IP，假设你的公网IP是：`116.222.222.222`

3. 配置文件

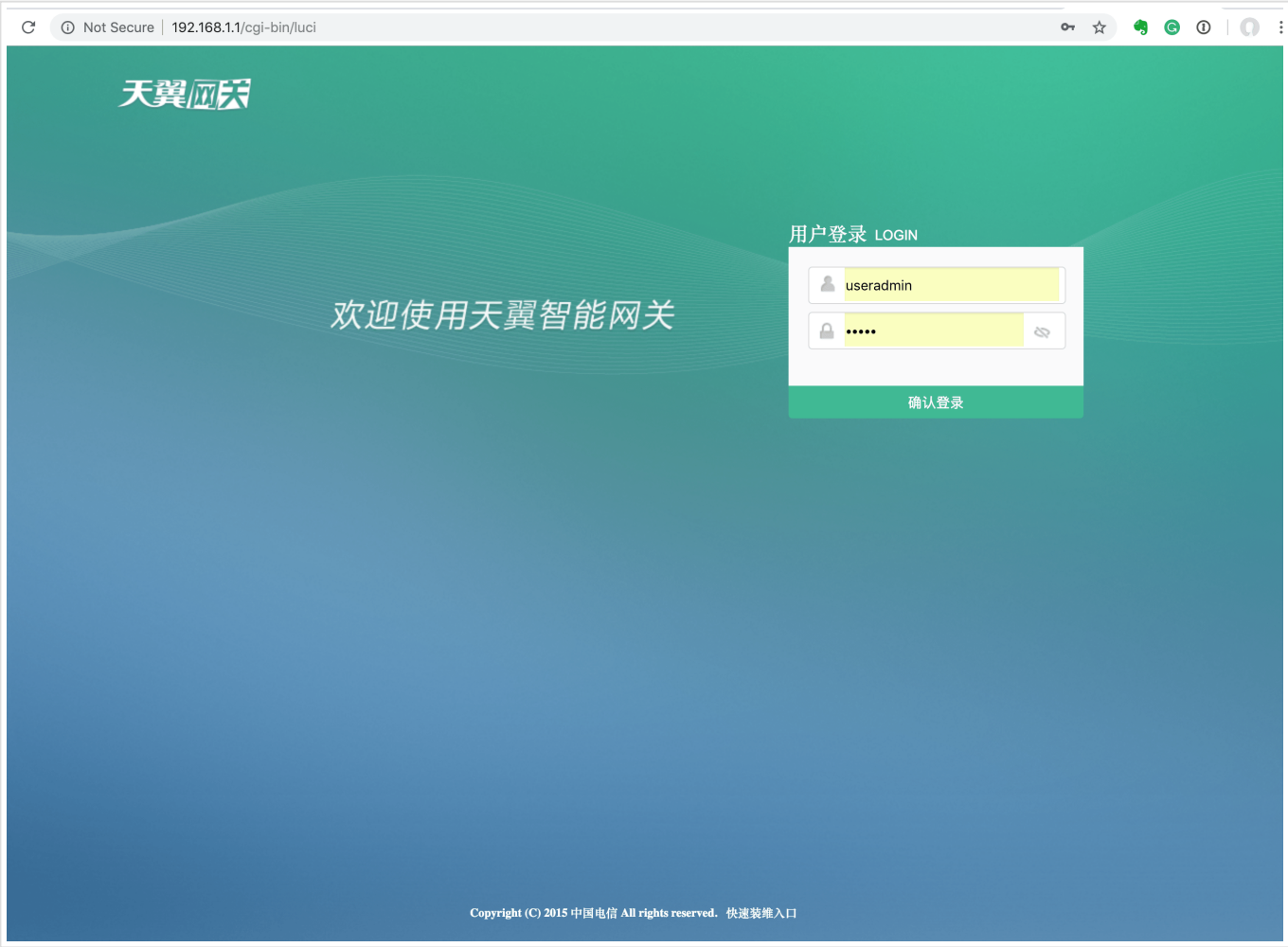
在 `~/openvpn` 下创建 `home.ovpn` 文件，并填写以下内容:

```
dev tap
proto udp

verb 3

ca ca.crt
cert my-client.crt
key my-client.key

client
remote-cert-tls server
remote 116.222.222.222 1194
```

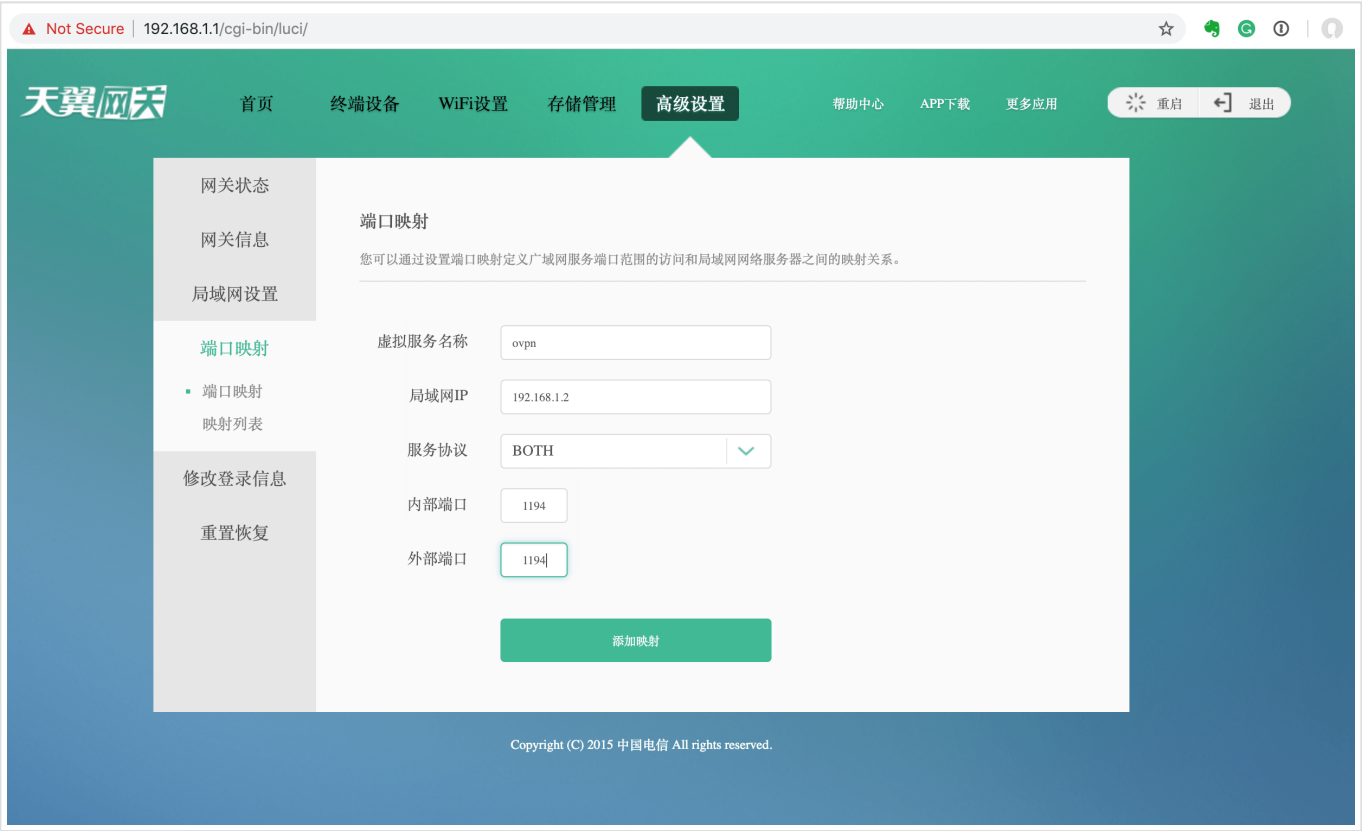
将该配置文件拖入Tunnelblick，以便于添加该ovpn。

4. 电信猫上做端口转发

通常来说电信的猫是拒绝 1194 这个端口的入口流量的，不过一般来说天翼网关是允许做非80端口的端口转发的，加入你的电信猫LAN口IP是 192.168.1.1 ，通过 http://192.168.1.1 访问天翼网关页面:



登录后，通过 高级设置 -> 端口映射 如下图添加映射，其中的 192.168.1.2 是咱们用于跑OpenVPN的OpenWRT路由器所被分配到的IP地址:

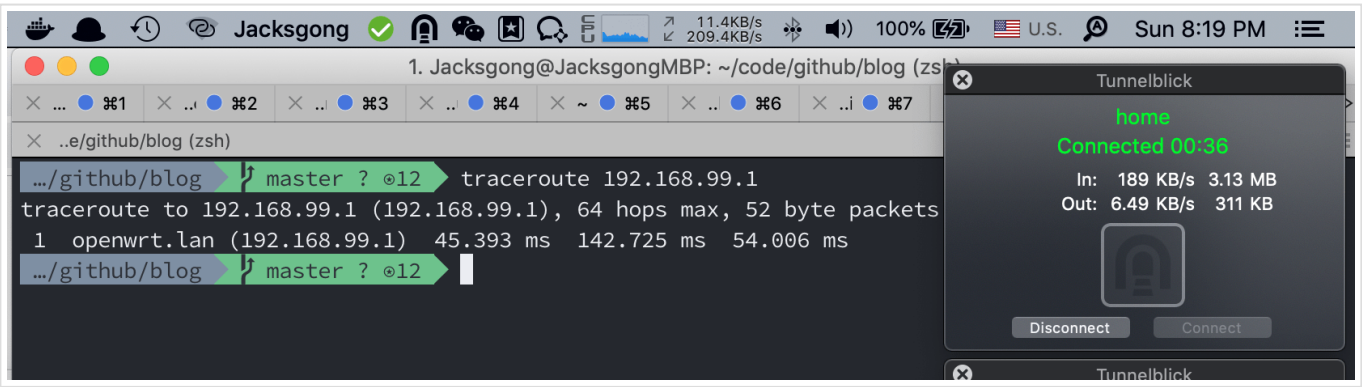


添加映射后，这边就 1194 端口上访问的流量就会自动被导到咱们的OpenWRT上，OpenWRT上的OpenVPN监听该端口的相关协议的流量，此时便可以正常访问了。

5. 建立连接

我们找到一个外网的环境，双击刚刚咱们添加的 home ，此时便建立连接了，建立连接后，虽然我们是在外网但是所有的 192.168.99.0/24 的流量都已经被路由到了我们家里的路由器上，我们可以简单的通过以下方法验证(下面的 192.168.99.1 是咱们路由器LAN口的IP):

```
1   traceroute 192.168.99.1
```



- OpenVPN Setup Guide for Beginners
- 利用openvpn远程连回家里openwrt路由器上内/外网。。。

欢迎关注Jacks Blog公众号，第一时间接收原创技术沉淀干货。

openwrt # cert # openvpn

流利说APM性能平台大盘工程实践
Mac 直播游戏，斗鱼直播，bilibili直播

What do you think?

1条回复

Upvote Funny Love

Surprised Angry Sad

评论 在线社区 1 登录

推荐 推文 分享 评分最高

加入讨论…

通过以下方式登录

或注册一个 DISQUS 帐号 ?

姓名

Kaiyuan • 4个月前

好像防火墙阻隔了访问 Lan 其他设备，我配置完之后无法访问局域网内的设备。

• 回复 • 分享 ›

在 JACKS BLOG 上还有

Kotlin Coroutines

1条评论 • 2年前

头像 chris zhao — 关于 5. 多个CoroutineContext进行+操作 commonpool + job

Jacks Blog

2条评论 • 1年前

头像 Jacks — 谢谢~

Android大项目开发

1条评论 • 3年前

头像 Even Wu — 你好，我想问

ARCore

5条评论 • 2年前

头像 Jacks — 好吧.....不过官方