正则搜索

置顶文章

Lotus 源码研究 05 - PC1 S...
Lotus 源码研究 02 - 1.10....
Lotus 1.10.0 FinalizeFaile...
Lotus 新手运维手册，持续...
矿工应该了解的几个 Filecoi...
从零开始搭建 Filecoin 主网...
Filecoin 运维(1) - 几个常...
Ceph-07 Ceph 运维常用指令
Ceph-06 Ceph 文件系统

最近发表

Ubuntu WIFI hardware-di...
历史总在重演，科技永远向前
Lotus 源码研究 05 - PC1 S...
Lotus 源码研究 04 - 小试...
Lotus 源码研究 03 - 源码...
Lotus 源码研究 03 - 爆块...
如何通过深度工作提高工作...
Lotus 源码研究 02 - 1.10....
Lotus 1.10.0 FinalizeFaile...
Lotus 网络升级 v1.10.0 - ...

文章分类 23

Database (6)
技术杂谈 (20)
Jekyll (1)
FunnyTools (7)
前端开发 (9)
Docker (3)
PHP (5)
随笔杂谈 (8)
Java (9)
SpringBoot (4)
读书笔记 (7)
区块链技术 (3)
C/C++ (1)
EOS (4)
以太坊 (7)
IPFS (5)
比特币 (2)
Filecoin (24)
Golang (2)
Sharding-JDBC (3)
Redis (1)
分布式存储 (11)

Ubuntu 搭建 OpenVPN 服务

2019-04-20 RockYang 16204

本文介绍如何在 Ubuntu 系统上搭建 OpenVPN 服务。

文章导读：

- 安装 OpenVPN
- 创建证书和秘钥
  - 修改并初始化环境变量
  - 创建秘钥
- 创建服务器端配置文件(server.conf)
- 配置内核和防火墙，启动服务端
- 创建客户端配置文件 client.ovpn （用于客户端软件使用）
- 在路由器上创建虚拟服务器
- 启动客户端
- 参考连接

首先说明一下我本机环境

- 操作系统：Ubuntu-18.04-LTS
- 内网 IP：192.168.1.110
- 外网 IP：14.153.76.90
- OpenVPN 版本：2.4.4

安装 OpenVPN

首先安装一些依赖，安装openssl和lzo，lzo用于压缩通讯数据加快传输速度

```
sudo apt-get install openssl libssl-dev
sudo apt-get install lzop
```

安装 OpenVPN 和 easy-rsa

```
sudo apt-get install openvpn
sudo apt-get install easy-rsa
```

创建证书和秘钥

安装完 easy-rsa 之后我们就可以开始创建 OpenVPN 服务所需要的秘钥了。

修改并初始化环境变量

```
sudo su
cd /usr/share/easy-rsa/
vim vars

# 修改注册信息，比如公司地址、公司名称、部门名称等。
export KEY_COUNTRY="CN"
```

Lotus-源码系列 (6)

</> 代码仓库

Herosphp框架

NKeditor 富文本编辑器

Mybatis-Kits

JS在线涂鸦小工具

电影票选座工具

Java联盟链

♥ 与我联系

友情链接

HerosPHP开发文档

RockYang 开源项目

原语云

```
export KEY_PROVINCE="GuangDong"
export KEY_CITY="ShenZhen"
export KEY_ORG="XJXH"
export KEY_EMAIL="rock@xjxh.io"
export KEY_OU="FuckItWhatever"
export KEY_NAME="EasyRSA"

# 使环境变量生效
source ./vars

# 添加 openssl 配置文档
cp openssl-1.0.0.cnf openssl.cnf
```

创建秘钥

```
# 清除keys目录下所有与证书相关的文件
# 下面步骤生成的证书和密钥都在/usr/share/easy-rsa/keys目录里
./clean-all

# 生成根证书ca.crt和根密钥ca.key（一路按回车即可）
./build-ca

# 为服务端生成证书和私钥，--batch 表示保持默认设置，无须回车确认
./build-key-server --batch server

# 为客户端生成证书和私钥
./build-key --batch client

# 创建迪菲·赫尔曼密钥，会生成dh2048.pem文件（生成过程比较慢，在此期间不要去中断它）
./build-dh

# 生成ta.key文件（防DDos攻击、UDP淹没等恶意攻击）
openvpn --genkey --secret keys/ta.key
```

创建服务器端配置文件(server.conf)

首先在 openvpn 的配置目录下新建一个 keys 目录

```
sudo mkdir -p /etc/openvpn/keys
```

然后，将需要用到的 openvpn 证书和密钥复制一份到刚创建好的 keys 目录中

```
cp /usr/share/easy-rsa/keys/{ca.crt,server.{crt,key},dh2048.pem,ta.key} /etc/openvp
```

复制一份服务器端配置文件模板 `server.conf` 到 `/etc/openvpn/`

```
gzip -d /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf /etc/openvpn/
```

编辑 `server.conf`

```
# 服务端口
port 1194

# 使用的传输协议
proto tcp
# 路由模式，桥接模式用dev tap
```

```
dev tun

# 证书路径
ca keys/ca.crt
cert keys/server.crt
key keys/server.key

dh keys/dh2048.pem

# 默认虚拟局域网网段，不要和实际的局域网冲突即可
server 10.8.0.0 255.255.255.0

ifconfig-pool-persist /var/log/openvpn/ipp.txt

# 192.168.1.0 是我的 OpenVPN 服务器所在在局域网的网段
# 如果你的局域网不是这个，那这里需要修改成你的网段
push "route 192.168.1.0 255.255.255.0"

# 如果客户端都使用相同的证书和密钥连接VPN，一定要打开这个选项，否则每个证书只允许一个人连接
duplicate-cn

# 这里如果设置了 tls-auth 则客户端也要设置，而且要跟服务器端对应，服务端为0，客户端则为 1
# 我这里注释掉了，因为我客户端没有使用 tls-auth
;tls-auth keys/ta.key 0 # This file is secret
;key-direction 0

# clients we want to allow.
max-clients 100

persist-key
persist-tun

status /var/log/openvpn/openvpn-status.log
log         /var/log/openvpn/openvpn.log
log-append  /var/log/openvpn/openvpn.log

verb 3

# 如果上面配置了传输方式为 TCP，则此处应该注释掉，否则会产生冲突
;explicit-exit-notify 1

# 这里配置使用用户名和密码登录的支持，可以取代使用秘钥和证书登录
auth-user-pass-verify /etc/openvpn/checkpsw.sh via-env
# 这里非常重要，如果你启用了该选项，你就只需要通过用户名和密码登录了
# 但是如果你注释了该选项，那你必须使用 用户名 + 密码 + 证书 才能登录成功，缺一不可。
;verify-client-cert none
username-as-common-name
script-security 3
```

如果你配置了使用用户名和密码登录，那么你需要创建登录验证脚本 `vim /etc/openvpn/checkpsw.sh`

```
#!/bin/sh
###########################################################
# checkpsw.sh (C) 2004 Mathias Sundman <mathias@openvpn.se>
#
# This script will authenticate OpenVPN users against
# a plain text file. The passfile should simply contain
# one row per user with the username first followed by
# one or more space(s) or tab(s) and then the password.
PASSFILE="/etc/openvpn/psw-file"
LOG_FILE="/etc/openvpn/openvpn-password.log"
```

```
TIME_STAMP=`date "+%Y-%m-%d %T"`
###############################################################
if [ ! -r "${PASSFILE}" ]; then
  echo "${TIME_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >
  exit 1
fi
CORRECT_PASSWORD=`awk '!/^;/&&!/^#/&&$1=="'${username}'"{print $2;exit}' ${PASSFILE
if [ "${CORRECT_PASSWORD}" = "" ]; then
  echo "${TIME_STAMP}: User does not exist: username=\"${username}\", password=\"${
  exit 1
fi
if [ "${password}" = "${CORRECT_PASSWORD}" ]; then
  echo "${TIME_STAMP}: Successful authentication: username=\"${username}\"." >> ${L
  exit 0
fi
echo "${TIME_STAMP}: Incorrect password: username=\"${username}\", password=\"${pas
exit 1
```

然后你还需要创建一个密码本文件 `vim /etc/openvpn/psw-file`，每一行一个用户，用户名和密码之间用空格隔开：

```
user1 pass1
user2 pass2
user3 pass3
```

至此服务端配置完成。

配置内核和防火墙，启动服务端

> 第一步，开启路由转发功能

```
sed -i '/net.ipv4.ip_forward/s/0/1/' /etc/sysctl.conf
sed -i '/net.ipv4.ip_forward/s/#//' /etc/sysctl.conf
sysctl -p
```

> 第二步，配置 `iptables`

```
iptables -I INPUT -p tcp --dport 1194 -m comment --comment "openvpn" -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j MASQUERADE
```

然后我们保存 iptables 设置，并在开机自动加载配置，初始化。这里可以通过 `iptables-persistent` 来快速实现

```
sudo apt-get install iptables-persistent
# 保存规则
sudo service netfilter-persistent save
```

下次开机启动的时候就可以看到 `iptables` 规则已经自动加载

```
$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:openvpn /* op
```

```
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

关闭 ufw 防火墙

```
ufw disable
```

> 第三步，启动 OpenVPN 服务

```
/etc/init.d/openvpn start
# 设置开机启动
systemctl enable openvpn@server
```

创建客户端配置文件 client.ovpn （用于客户端软件使用）

首先复制一份 `client.conf` 模板命名为 `client.ovpn`

```
mkdir ~/openvpn-client
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/openvpn-client
```

然后修改 client.ovpn, `vim /etc/openvpn/client.ovpn`

```
client
# 这里设置跟服务端一样
dev tun
proto tcp

# OpenVPN 服务端 IP 和端口
remote 14.153.76.90 1194

resolv-retry infinite

nobind

persist-key
persist-tun

# 这里设置证书和秘钥
ca ca.crt
cert client.crt
key client.key

remote-cert-tls server

# 如果服务端使用 tls-auth，则这里也要启用
;tls-auth ta.key 1

comp-lzo

verb 3

# 用来存放用户名和密码的文件路径，这样在连接的时候就不需要手动输入用户名密码了
```

```
auth-user-pass pass.txt
```

修改好客户端配置文档之后，接下来需要把证书文件也一起拷贝到 `~/openvpn-client` 文件夹中：

```
cp /ect/openvpn/keys/ca.crt ~/openvpn-client
cp /usr/share/easy-rsa/keys/client.crt ~/openvpn-client
cp /usr/share/easy-rsa/keys/client.key ~/openvpn-client
```

然后还需要在 `openvpn-client` 目录中新建 `pass.txt` 文件，第一行写用户名，第二行写密码：

```
user1
pass1
```

在路由器上创建虚拟服务器

这一步非常重要！！！ 虽然你的 OpenVPN 服务已经正常启动了，但是你会发现你还是无法连接它，因为外界还无法跟你的 OpenVPN 服务器通信， **路由器默认是会拦截并丢弃外侧网络发起的主动连接，也就是说路由器默认是只能让你主动出去，而不允许网面的主动请求进来。** 原因很简单，这样能保证你内部局域网主机的安全。 **路由器两端连接的是不同的网络，而不同网络之间是不能相互通信的。**

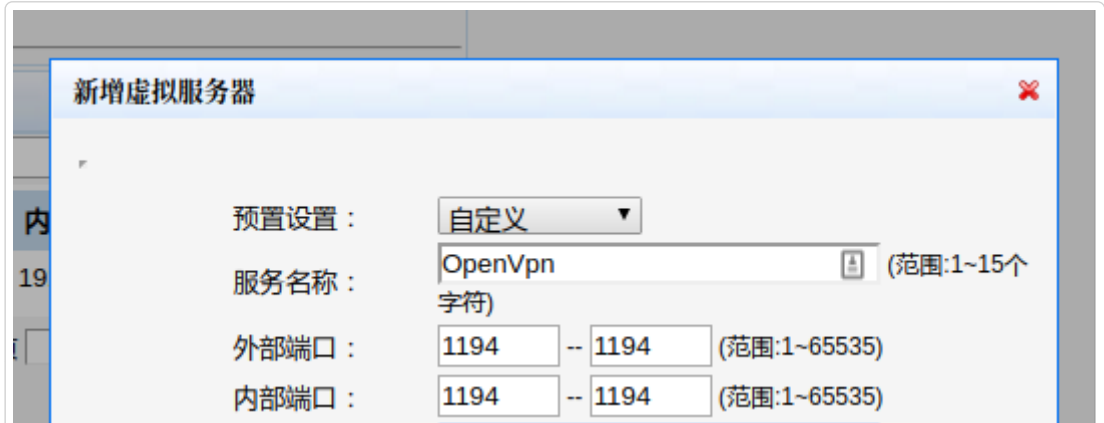
> 因为对于外部的主机来说，我们内部局域网所有主机都是共享一个公网 IP，比如目前我所在的局域网的共享公网 IP 为 14.153.76.90，那么此时外部主机向 14.153.76.90 发送一条消息，路由器应该怎么处理呢，因为此时路由器并不知道你这条消息要发给谁，是发给 192.168.0.110 还是 192.168.0.111 呢？它无法判断，所以只能把这个消息包丢弃。

所以我们需要在路由器上配置一个虚拟服务器，告诉路由器，如果外部有请求进来，它应该把请求转给内网的哪台主机。

下面我以 H3c 路由器为例，演示如何添加虚拟服务器。

首先登录路由器管理界面，进入 “高级设置” -> “地址转换” -> “虚拟服务器”

然后点击 “新增” 按钮，在弹出的对话框中按照下图填写好就 OK 了。



此次，大功告成，你的 VPN 服务器可以正常工作了。

启动客户端

启动客户端很简单，只需要把上一步我们创建的 `openvpn-client` 文件夹拷贝到客户端机器，然后执行下面的命令：

```
cd openvpn-client
sudo openvpn --config client.ovpn
```

如果你看到输出类似下面的日志，则说明你已经成功连接上了 VPN，你可以直接访问 VPN 服务器所在的网络了。

```
Sat Apr 20 14:30:34 2019 /sbin/ip link set dev tun0 up mtu 1500
Sat Apr 20 14:30:34 2019 /sbin/ip addr add dev tun0 local 10.8.0.6 peer 10.8.0.5
Sat Apr 20 14:30:34 2019 /sbin/ip route add 192.168.0.0/24 via 10.8.0.5
Sat Apr 20 14:30:34 2019 /sbin/ip route add 10.8.0.1/32 via 10.8.0.5
Sat Apr 20 14:30:34 2019 Initialization Sequence Completed
```

参考连接

- https://olei.me/181/
- https://www.cnblogs.com/EasonJim/p/8339600.html

技术杂谈 20　OpenVPN 1　Ubuntu 3