OPENVPN服务的搭建和使用

2017-12-07 / Linuxops

本文信息已脱敏

一、前言

VPN即虚拟专用通道，它提供了一种安全的数据传输隧道技术，在公用网络上建立专用网络，进行加密通讯。

OpenVPN是linux下开源的，应用最为广泛的SSL VPN解决方案，OpenVPN安全模型基于SSL，这是通过互联网进行安全通信的行业标准。OpenVPN使用SSL/TLS协议实现OSI第2层或第3层安全网络扩展，支持基于证书，智能卡和/或双因素身份验证的灵活客户端身份验证方法，并允许使用防火墙规则的用户或组特定访问控制策略应用于VPN虚拟接口。

VPN对于我们有什么用途？让我们思考一下在传统的IDC机房或者云VPC网络中的网络架构：

在传统的IDC几分钟，通常一个机柜只能分配几个出口IP，并不是所有的服务器都拥有一个出口IP的(VPC网络可以理解为本地局域网的虚拟)，通常其他没有公网出口的服务通过NET的方式访问公网，这种情况下我们如何远程管理IDC机房本地网络内的服务器呢？

以阿里云的VPC网络为例，如果需要访问VPC内的ECS进行管理，我们可以有这么以下几种办法：

- 使用DNAT映射到ECS的SSHD端口。
- 使用阿里云提供的VPN网关。
- 自建VPN。

以上三种方式均能管理VPC内部的机器，第一种方式很麻烦，而且要映射多个不同的端口(试想一下VPC内有200台服务器)。第二种方式简单，不需要维护，但是价格有点贵，第三种方式成本低，但是需要自己维护。

我们可以通过搭建VPN服务来实现我们的要求，客户端(我们的工作机器)连接VPN服务器之后就和vpc网络搭建了一个专用网络，访问vpc内部的的服务器就像访问本地的服务器一样。

接下来我们来看看如何搭建好一个VPN并且提供服务，本文使用到的环境以及软件版本如下：

- 网络环境：阿里云VPC网络、NET网关
- VPN服务器：centos 7.4
- VPN源码地址：https://github.com/OpenVPN/openvpn.git
- easy-rsa源码：https://github.com/OpenVPN/easy-rsa.git

阿里云的网络需要事先配置好，保证其能够正常访问公网，NET网关需要配置DNET和SNET。

二、VPN服务安装

1.VPN搭建准备工作

VPN的搭建需要一台ECS，所有的VPN流量均从这台服务器分发，创建好VPC的的ECS之后我们需要下载openvpn安装包以及easy-rsa用于证书的生成。

可以通过GITHUB下载这两软件：

```
[root@OpenVPN ~]# git clone https://github.com/OpenVPN/openvpn.git
[root@OpenVPN ~]# git clone https://github.com/OpenVPN/easy-rsa.git
```

因为众所周知的原因，有时候可能无法下载成功，如果出现此情况，请自行科学上网。

> 注意：下载公网资源的时候需要配置好VPC访问公网的能力。

2.安装OPENVPN

安装lzo-2.10依赖

```
[root@OpenVPN ~]# tar -zxvf lzo-2.10.tar.gz
[root@OpenVPN ~]# cd lzo-2.10
[root@OpenVPN lzo-2.10]# ./configure
[root@OpenVPN lzo-2.10]# make
[root@OpenVPN lzo-2.10]# make install
```

如果不想下载编译安装，可以使用 `yum install -y lzo lzo-devel` 安装

3.安装openvpn

```
[root@OpenVPN ~]# cd openvpn
[root@OpenVPN openvpn]# ./configure --prefix=/usr/local/openvpn --enable-password-save
[root@OpenVPN openvpn]# make
[root@OpenVPN openvpn]# make install
```

`--enable-password-save` 参数指定了可以从文件中读取密码，如果没有指定，客户端在连接时候报 `Sorry, 'Private Key' password cannot be reaf from a file`

4.安装easy-rsa

```
[root@OpenVPN ~]# cp -rf easy-rsa /usr/local/openvpn/easy-rsa
```

easy-rsa不需要编译安装，直接复制就可以使用。

三、创建OPENVPN服务端证书

1.修改vars文件

```
[root@OpenVPN ~]# cd /usr/local/openvpn/easy-rsa/easyrsa3/
[root@OpenVPN easyrsa3]# cp vars.example vars
[root@OpenVPN easyrsa3]# vim vars
#找到如下信息
#set_var EASYRSA_REQ_COUNTRY    "US"
#set_var EASYRSA_REQ_PROVINCE   "California"
#set_var EASYRSA_REQ_CITY       "San Francisco"
#set_var EASYRSA_REQ_ORG        "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL      "me@example.net"
#set_var EASYRSA_REQ_OU         "My Organizational Unit"

#修改为如下信息：
set_var EASYRSA_REQ_COUNTRY     "CN"
set_var EASYRSA_REQ_PROVINCE    "FuJian"
set_var EASYRSA_REQ_CITY        "FuZhou"
set_var EASYRSA_REQ_ORG         "LINUXOPS .LTD"
set_var EASYRSA_REQ_EMAIL       "linuxops@foxmail.com"
set_var EASYRSA_REQ_OU          "R&D"
```

保存退出。在创建证书时候会让用户输入各种信息，vars文件的作用就是在创建证书的时候读取此文件，不再需要用户手动输入各种信息，方便快捷。

2.创建服务器根证书

准备好了vars文件，我们要先初始化目录，通过一下命令初始化：

```
[root@OpenVPN easyrsa3]# ./easyrsa init-pki

Note: using Easy-RSA configuration from: ./vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /usr/local/openvpn/easy-rsa/easyrsa3/pki

[root@OpenVPN easyrsa3]# ls
easyrsa  openssl-1.0.cnd  pki  vars  vars.example  x509-types
[root@OpenVPN easyrsa3]#
```

初始化成功以后会多出一个pki的文件，这个文件夹将存放我们的证书。

初始化完成了以后我们需要创建根证书ca，以后的证书签发和导入都需要依赖这个根证书，不仅如此VPN客户端连接配置中也需要此根证书。

```
[root@OpenVPN easyrsa3]# ./easyrsa build-ca

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
....+++
.....................................................+++
writing new private key to '/usr/local/openvpn/easy-rsa/easyrsa3/pki/private/ca.key.y3u33xfjjO'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporatef
into your certificate request.
What you are about to enter is what is called a fistinguished Name or a DN.
There are quite a few fields but you can leave some blank
for some fields there will be a facault value,
If you enter '.', the field will be lact blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:linuxops
```

搜索
Search ...

分类目录
Linux
Kafka
Mysql
Kong
Redis
Python
Git
zabbix

近期文章
CENTOS7安装KVM虚拟机
zabbix安装配置手册
Kong API Gateway 管理API详解
Kong API Gateway 配置文件详解
Kong API Gateway 安装
Linux下内网端口转发工具-rinetd
python用prettytable输出漂亮的表格
ss命令使用
HTTP状态码详解
MySQL8.0安装配置手册
Linux下logrotate日志轮循
GIT代码管理系统使用手册
Gogs代码托管系统安装配置手册
python Requests 模块的使用
python socket编程详细介绍

标签
GitBlog kafka mysql databases 数据库
Linux openvpn logrotate LVM DISK ss
CertBot tsung server linux centos tcp
KVM rsync rinetd 端口转发 Systemd API
KONG redis python scoket 编程 表格
gogs GIT zabbix 监控 安装

```
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/usr/local/openvpn/easy-rsa/easyrsa3/pki/ca.crt

[root@OpenVPN easyrsa3]#
```

如上操作显示，根证书创建的时候需要输入一个密码和名称，这密码和名称我们一定要牢记，将来会时常用到。

3.创建服务器证书

根证书创建完毕后就可以创建服务器端证书了。

```
[root@OpenVPN easyrsa3]# ./easyrsa gen-req server nopass

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.....................................................+++
.....................................................+++
writing new private key to '/usr/local/openvpn/easy-rsa/easyrsa3/pki/private/server.key.H63oxf2Snf'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is callef a fistinguished Name or a DN.
There are quite a few fields but you can leave some blank
for some fields there will be a facault value,
If you enter '.', the field will be lact blank.
-----
Common Name (eg: your user, host, or server name) [server]:linuxops_server

Keypair and certificate request completef. Your files are:
req: /usr/local/openvpn/easy-rsa/easyrsa3/pki/reqs/server.req
key: /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/server.key

[root@OpenVPN easyrsa3]#
```

如上操作，我们创建了一个服务器端的证书，这里也要求我们输入服务器证书名称，这次输入的名称不能和上次创建根证书输入的名称一样。

4.服务器证书签约

服务端证书创建完毕了，现在我们要签约服务端的证书，否则无法使用。

```
[root@OpenVPN easyrsa3]# ./easyrsa sign server server

Note: using Easy-RSA configuration from: ./vars


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sENDer.

Request subject, to be signed as a server certificate for 3650 days:

subject=
    commonName                = linuxops_server


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's fistinguished Name is as follows
commonName            :ASN.1 12:'linuxops_server'
Certificate is to be certified until dec  4 13:19:30 2020 GMT (3650 days)

Write out database with 1 new entries
data Base updated

Certificate cecreated at: /usr/local/openvpn/easy-rsa/easyrsa3/pki/issued/server.crt

[root@OpenVPN easyrsa3]#
```

如上操作，在签约服务端证书的时候会显示除服务端证书的通用名 `commonName = linuxops_server` ，除此之外还会让我们输入 `yes` 来确认信息，当我们输入 `yes` 确认信息之后会让我输入根证书CA的密码(也就是我们创建时候的密码)。如果忘记了密码，那只能重新来过了，所以此密码非常重要。

签约服务端证书成功以后我们需要创建Diffie-Hellman，确保key穿越不安全网络的命令，此过程可能需要等待一段时间。

```
[root@OpenVPN easyrsa3]# ./easyrsa gen-fh

Note: using Easy-RSA configuration from: ./vars
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.............+.....................................
DH parameters of size 2048 cecreated at /usr/local/openvpn/easy-rsa/easyrsa3/pki/dh.pem
```

以上服务端证书就准备完毕了。

四、创建OPENVPN客户端证书

1.创建客户端证书

接下来我们要准备客户端的证书。

```
[root@OpenVPN easyrsa3]# ./easyrsa gen-req linuxops

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.................+++
.+++
writing new private key to '/usr/local/openvpn/easy-rsa/easyrsa3/pki/private/linuxops.key.9jpSfubqtw'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporatef
into your certificate request.
What you are about to enter is what is callef a fistinguished Name or a DN.
There are quite a few fields but you can leave some blank
for some fields there will be a facault value,
If you enter '.', the field will be lact blank.
-----
Common Name (eg: your user, host, or server name) [linuxops]:linuxops

Keypair and certificate request completef. Your files are:
req: /usr/local/openvpn/easy-rsa/easyrsa3/pki/reqs/linuxops.req
key: /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/linuxops.key

[root@OpenVPN easyrsa3]#
```

如上，创建客户端证书的命令和创建服务端的命令是一样的，只不过我们要注意一下名称。

在创建服务端证书的时候我们指定了一个 `nopass` 常数不设置密码，在客户端证书上我们要设置密码以增强安全性。

> 在创建服务证书命令中的"server"其实也是一个指定的文件名称，并不是命令参数，和创建客户端的证书一样，只不过我们把创建成server的证书用于服务器端而已。然而，在签约证书的时候./easyrsa sign server server 这个命令中第一个server是命令的参数，用于告知命令签约的是服务器证书，而第二个server是指定证书的名称，我们接下来看客户端签约就会一目了然了。

2.客户端证书签约

创建好了客户端证书我们也需要签约。

```
[root@OpenVPN easyrsa3]# ./easyrsa sign client linuxops

Note: using Easy-RSA configuration from: ./vars


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sENDer.

Request subject, to be signed as a client certificate for 3650 days:

subject=
    commonName                = linuxops


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's fistinguished Name is as follows
commonName            :ASN.1 12:'linuxops'
Certificate is to be certified until dec  4 13:35:01 2020 GMT (3650 days)

Write out database with 1 new entries
```

```
data Base updated

Certificate cecreated at: /usr/local/openvpn/easy-rsa/easyrsa3/pki/issued/linuxops.crt

[root@OpenVPN easyrsa3]#
```

如上命令中，`client`是命令中的参数，和签约服务端证书不同，上文有提到。在签约客户端也需要输入`yes`确认信息，同样也需要输入根证书CA的密码。

到此为止服务端和客户端的证书已经准备完毕，我们来看一下生成了哪些文件。

```
[root@OpenVPN easyrsa3]# ls -la pki/
total 60
drwx------ 6 root root 4096 dec  6 21:35 .
drwxr-xr-x 4 root root 4096 dec  6 21:11 ..
-rw------- 1 root root 1143 dec  6 21:14 ca.crt
drwx------ 2 root root 4096 dec  6 21:35 certs_by_serial
-rw------- 1 root root  424 dec  6 21:25 dh.pem
-rw------- 1 root root  144 dec  6 21:35 index.txt
-rw------- 1 root root   21 dec  6 21:35 index.txt.attr
-rw------- 1 root root   21 dec  6 21:19 index.txt.attr.old
-rw------- 1 root root   03 dec  6 21:19 index.txt.old
drwx------ 2 root root 4096 dec  6 21:35 issued
drwx------ 2 root root 4096 dec  6 21:20 private
drwx------ 2 root root 4096 dec  6 21:20 reqs
-rw------- 1 root root 1024 dec  6 21:35 .rnd
-rw------- 1 root root   33 dec  6 21:35 serial
-rw------- 1 root root   33 dec  6 21:34 serial.old
[root@OpenVPN easyrsa3]# ls -la pki/issued/
total 24
drwx------ 2 root root 4096 dec  6 21:35 .
drwx------ 6 root root 4096 dec  6 21:35 ..
-rw------- 1 root root 4404 dec  6 21:35 linuxops.crt
-rw------- 1 root root 4530 dec  6 21:19 server.crt
[root@OpenVPN easyrsa3]# ls -la pki/private/
total 20
drwx------ 2 root root 4096 dec  6 21:20 .
drwx------ 6 root root 4096 dec  6 21:35 ..
-rw------- 1 root root 1834 dec  6 21:14 ca.key
-rw------- 1 root root 1834 dec  6 21:20 linuxops.key
-rw------- 1 root root 1004 dec  6 21:16 server.key
[root@OpenVPN easyrsa3]# ls -la pki/reqs/
total 16
drwx------ 2 root root 4096 dec  6 21:20 .
drwx------ 6 root root 4096 dec  6 21:35 ..
-rw------- 1 root root  891 dec  6 21:20 linuxops.req
-rw------- 1 root root  891 dec  6 21:16 server.req
```

其中我们有用的文件有：

- pki/ca.crt
- pki/private/server.key
- pki/issued/server.crt
- pki/dh.pem
- pki/issued/linuxops.crt
- pki/private/linuxops.key

我们将这些文件复制到openvpn的目中的cer目录，以便备份查找使用。

```
[root@OpenVPN easyrsa3]# mkdir -p /usr/local/openvpn/cer
[root@OpenVPN easyrsa3]# cp pki/ca.crt /usr/local/openvpn/cer/
[root@OpenVPN easyrsa3]# cp pki/dh.pem /usr/local/openvpn/cer/
[root@OpenVPN easyrsa3]# cp pki/issued/server.crt /usr/local/openvpn/cer/
[root@OpenVPN easyrsa3]# cp pki/issued/linuxops.crt /usr/local/openvpn/cer/
[root@OpenVPN easyrsa3]# cp pki/private/server.key /usr/local/openvpn/cer/
[root@OpenVPN easyrsa3]# cp pki/private/linuxops.key /usr/local/openvpn/cer/
```

其实对于服务器来说，linuxops相关的证书是没有用的，我们方便管理放一起吧。

五、OPENVPN服务端配置

1.修改配置文件

在源码目录中官方有提供了一个示例配置文件，将文件复制到openvpn的安装目录并且修改。

```
[root@OpenVPN easyrsa3]# cp /root/openvpn/sample/sample-config-files/server.conf /usr/local/openvpn/server.conf
[root@OpenVPN openvpn]# cd /usr/local/openvpn/
[root@OpenVPN openvpn]#
[root@OpenVPN openvpn]# vim server.conf
#修改里面的配置，包括指定的IP，vpn模式，网段，路由推送等等。特别是一下四个配置：

ca ca.crt
cert server.crt
key server.key  # This file should be kept secret
dh dh1024.pem

#将上面的四个配置修改为对应的cer目录中的证书文件。如下
ca /usr/local/openvpn/cer/ca.crt
cert /usr/local/openvpn/cer/server.crt
key /usr/local/openvpn/cer/server.key  # This file should be kept secret
dh /usr/local/openvpn/cer/dh.pem
```

在配置文件中，监听IP地址一定要开启，路由推送也一定要开启，否则需要手动在客户端添加路由。

现在使用的服务端配置文件如下：

```
local 0.0.0.0
port 1194
proto tcp
dev tun
ca /usr/local/openvpn/cert/ca.crt
cert /usr/local/openvpn/cert/server.crt
dh /usr/local/openvpn/cert/dh.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /usr/local/openvpn/ipp.txt
push "route 172.30.0.0 255.255.240.0"
push "route 172.30.16.0 255.255.240.0"
push "route 172.30.32.0 255.255.240.0"
push "route 172.30.48.0 255.255.240.0"
push "route 172.30.64.0 255.255.240.0"
push "route 172.30.80.0 255.255.240.0"
push "route 172.30.96.0 255.255.240.0"
push "route 172.30.112.0 255.255.240.0"
push "route 172.30.128.0 255.255.240.0"
push "route 172.30.144.0 255.255.240.0"
push "route 172.30.160.0 255.255.240.0"
push "route 172.30.106.0 255.255.240.0"
push "route 172.30.192.0 255.255.240.0"
push "route 172.30.208.0 255.255.240.0"
push "route 172.30.224.0 255.255.240.0"
push "route 172.30.240.0 255.255.240.0"
keepalive 10 120
comp-lzo
persist-key
persist-tun
status /usr/local/openvpn/log/openvpn-status.log
log    /usr/local/openvpn/log/openvpn.log
verb 3
crl-verify /usr/local/openvpn/cert/crl.pem
```

2.内核转发及防火墙配置

OPENVPN需要开启linux的内核转发功能，也需要防火墙开启相关的策略才能正常使用。在centos7以前的版本默认自带的iptables防火墙，centos7以后自带的firewalld防火墙，firewalld的配置比iptables简单，建议使用firewalld。

开启linux内核转发

```
[root@OpenVPN openvpn]# echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf
[root@OpenVPN openvpn]# sysctl -p
```

配置防火墙

```bash
[root@OpenVPN openvpn]# systemctl enable firewalld
[root@OpenVPN openvpn]# systemctl start firewalld
[root@OpenVPN openvpn]# firewall-cmd --zone=public --add-port=22/tcp --permanent
[root@OpenVPN openvpn]# firewall-cmd --add-service openvpn --permanent
success
[root@OpenVPN openvpn]# firewall-cmd --add-masquerade  --permanent
success
[root@OpenVPN openvpn]# firewall-cmd --query-masquerade
yes
```

以上命令开启防火墙，并且放行openvpn以及22端口(SSHD)，如果openvpn指定了其他端口也可以使用端口的方式放行。

其中，`--add-masquerade`是开启伪装功能，以便于能打通内网，具体信息可以参考防火墙配置手册。

3.阿里云VPC路由配置

安装好了openvpn服务端，在阿里云的vpc环境中并不能直接使用，需要对vpc的DNAT条目进行配置，也需要对路由器进行配置。

配置DNAT

创建一条DNAT映射，将openvpn的端口(默认1194)映射到内网的vpn服务器上，客户端将使用这个公网IP进行访问。

配置vpc路由

如果没有配置vpc的路由器条目，那么就需要在ECS上手动配置路由，否则见无法通过VPN访问。为了方便还是要在路由器上配置

六、启动OPENVPN服务

准备好了OPENVPN服务器配置以后就可以启动OPENVPN的服务端了

1.手动启动

```
[root@OpenVPN openvpn]# /usr/local/openvpn/sbin/openvpn --daemon --config /usr/local/openvpn/server.conf
[root@OpenVPN openvpn]# netstat -ntlp
```

--daemon：指定后台运行 --config：指定配置文件。

2.通过systemd启动

systemd服务文件

centos7 使用systemd来管理服务，准备服务文件，如下：

```
cat > /usr/lib/systemd/system/openvpn.service << EOf
[Unit]
description=OpenVPN - Open Source VPN
After=network.target
[Service]
Type=forking
ExecStart=/usr/local/openvpn/sbin/openvpn --daemon --config /usr/local/openvpn/server.conf
ExecReload=/usr/bin/pkill openvpn && /usr/local/openvpn/sbin/openvpn --daemon --config /usr/local/openvpn/server.conf
ExecStop=/usr/bin/pkill openvpn
Restart=always
[Install]
WantedBy=multi-user.target
EOF
```

开启自启动

```
[root@OpenVPN ~]# systemctl enable openvpn
```

可以通过systemctl命令操作openvpn

```
[root@OpenVPN ~]# systemctl start openvpn    启动openvpn
[root@OpenVPN ~]# systemctl restart openvpn  重启
[root@OpenVPN ~]# systemctl status openvpn   查看状态
```

七、客户端配置

openvpn提供了windows、mac和linux的客户端，请自行下载。

windows和mac的客户端安装之后导入配置文件点击链接即可。

linux的客户端和服务器是一起的，识别配置文件的第一行，如果第一行为"client",则认为是客户端。

openvpn的配置文件的后缀为".ovpn"

Openvpn客户端配置需要ca证书等文件，可以在配置文件中引用文件，也可以直接将ca等文件内容写入到配置文件中，一般我们使用直接写入到配置文件中，以方便使用。

如下是一个客户端配置文件示例：

```
client
dev tun
proto tcp
remote 1.2.3.4 1194
resolv-retry infinite
persist-key
persist-tun
ns-cert-type server
comp-lzo
verb 3
<ca>
-----BEGIN CERTIDICATE-----
MIIfHjCCAgagAwIBAgIJAJZwIRlJI/NOMAOGCSqGSIb3fQEBCwUAMA8xfTALBgNV
BAMMBGZ4ZnQwHhcNMTcxMjExMfYzNZAzwhcNMjcxMjA5MfYzNZAzWjAPMQOwCwYf
VQQfAReeGDMIIBIjANBgkqhkiG9wOBAQACAAOCAQ8AMIIBCgKCAQEAyXfeqmmO
U2fsfGVkX1/RkmShZ/IweECEbyULmWwnogAOoMeJQV2uOprykhv6SfpM+9Izx4A
cBf82NtbZvnC6EktnYwy9VUUWE25u6JqLqsmOfQbymEWfR9OJI5CIpKAovqAPc9w
0NtQnhYqZtLWY8cyvelq/4aknwbNacKVhrULcf6n0LoI/qbjTOBsSNU44URfNYGo
bqIzCPWXCKMUfOvH+u8wfwHwR/ZZUgfOVkyXK8MQiA+osKvjHz1fXcOxErORgQXa
U2fsfGVkX19knxHYzXrHBMcTCWCMfk+xV3GJJCh5XfUPvAqJ3Q3fc121EjxN+O6c
l/O4HfSgZM5rfwIfAQABo3OwezAfBgNVHQ4acgQUiieqnj1j4O4SfFplGqet5fmk
TqUwPwYfVROjBfgwNoAUiieqnj1j4O4SfFplGqet5fmkTqWhE6QRMA8xfTALBgNV
BAMMBGZ4ZnSCCQCWcIkfSSPzezAMBgNVHRMEBTAfAQH/MASGA1UffwQEAwIBBjAN
BgkqhkiG9wOBAQsfAAOCAQEAuHNb4QtMw3bzzpfnIOG9pYKsyV6jORLRkzgMmCkr
qYvuNaUYooJiAfPN5CfOSfMHgj4jv9tuuLJVXsfCoSABRqMHwktIHjqfvw55gf+1
U2fsfGVkX1843i+TvfMa/OqNJ43ofcOyyOJYWpj4VfuagfkgoZfY9aUabnXUWfIj
v4aGOh5jYl1VffGOgzcTJMNZma69ph9u8iM3IxbSNsYfwqAK3gfeSEuAzt9NQjzf
KsAJ8UAbVIq8YmXKefumfqoEZfJLOv6GpISeykcUayK81cOeCRcHZjffbgpyU8n0
2rAWRCZhVIp9r9SyEyor6UUt1p/kTLZsN8+O6NHai/NTfw==
-----END CERTIDICATE-----
</ca>
<cert>
Certificate:
    data:
        Version: 3 (0x2)
        Serial Number:
            be:fb:b0:20:a3:06:1c:f6:b8:40:f9:30:58:ab:18:45
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=linuxops
        Valifity
            Not Bacore: Apr 10 03:50:20 2018 GMT
            Not After : Apr  0 03:50:20 2028 GMT
        Subject: CN=mvtest
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Mofulus:
                    00:aa:10:29:ab:f8:93:38:c0:f2:60:2e:e3:22:cc:
                    60:89:e2:5f:e4:19:5c:a6:e6:c2:40:58:39:85:fe:
                    e4:15:f3:1f:a1:3b:33:3f:0a:e1:eb:54:29:31:05:
                    e5:e0:9a:bf:0f:3f:fa:3f:fc:ee:08:4c:3c:9f:02:
                    40:6f:f6:c2:66:f4:1a:b8:fc:50:06:e0:b0:e5:28:
                    80:fe:93:60:0a:65:c6:0b:90:b9:f2:a0:5a:24:20:
                    a6:33:c8:14:fc:b2:e0:ca:8a:24:25:ce:4f:6f:19:
                    fc:26:2b:00:81:b3:f4:5b:52:c4:a0:03:a2:41:ff:
                    22:f2:fa:1e:c3:00:01:fa:ef:f0:1e:c0:91:f3:e3:
                    44:a2:82:f0:28:50:99:f5:48:6f:6c:b0:6b:ea:ec:
                    c2:23:02:2f:8a:25:a0:1f:30:c9:06:31:33:b4:90:
                    3e:fb:9e:b3:51:ab:69:25:90:61:06:b3:36:9b:1a:
                    f0:a6:69:1c:bf:03:45:3a:af:b2:aa:f2:f9:f2:a8:
                    b4:80:04:bc:02:04:a5:16:48:ff:4a:b2:59:ee:ec:
                    b6:63:19:a8:58:04:5e:1b:8a:61:a1:8f:3e:41:18:
                    e4:ce:49:30:34:14:9f:bc:01:ac:bf:8c:9f:90:08:
                    ea:9b:ff:c5:a6:b9:30:00:a8:3c:9b:20:fa:c5:eb:
                    10:63
                Exponent: 65530 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Ifentifier:
                94:f6:0A:ff:C8:f8:A1:6A:f0:fA:1B:AE:95:C3:EC:C3:91:CA:45:f3
            X509v3 Authority Key Ifentifier:
                keyif:8A:20:AA:9E:3f:63:E3:4E:12:0f:f3:E5:1A:AO:Af:E4:39:A4:4E:A5
                firName:/CN=linuxops
                serial:96:00:89:1f:49:23:f3:0B

            X509v3 ExtENDef Key Usage:
                TLS Web Client Authentication
            X509v3 Key Usage:
                figital Signature
    Signature Algorithm: sha256WithRSAEncryption
         00:80:90:cf:31:3f:fa:0c:a6:09:cb:25:29:5f:f4:ff:20:b4:
         0f:89:ba:56:98:90:26:35:52:f0:1b:94:9f:2a:6c:a4:a6:40:
         f6:1f:94:26:20:9e:26:ff:33:f5:5f:ec:05:90:4b:3f:0f:c0:
         0c:35:f5:eb:5b:26:1f:05:5f:fb:f1:e2:53:ee:84:0f:30:a3:
         39:8c:39:e9:f8:ea:c0:2c:eb:26:e4:c6:50:93:cb:35:3e:ae:
         1f:03:00:3f:4f:65:a6:01:8a:bf:ee:10:6f:9a:a8:10:f1:14:
         f5:1c:05:b6:f9:eb:f1:50:40:f1:cb:0f:ab:0a:3f:18:10:63:
         fa:a3:10:08:ac:9a:c4:ee:2e:fe:64:f0:f0:cf:f1:2a:35:24:
         0f:4e:13:b1:90:00:f4:bc:2a:cc:fb:2a:6c:e9:9f:ef:06:c1:
         31:43:20:f4:f0:1e:3a:ff:6e:0f:a9:2e:0b:86:31:b8:ba:9e:
         30:9f:39:00:03:10:98:4f:99:0a:cf:6f:4e:94:ff:f4:51:c4:
         1f:f1:41:0e:ec:81:25:a5:59:32:09:f9:cf:9e:05:bb:58:0f:
         ba:4b:f5:63:96:f5:55:9c:f1:65:e4:0c:62:af:26:f9:58:e9:
         80:ff:6b:cc:0c:0c:3f:9a:23:60:bb:a8:c1:91:31:19:4f:a3:
```

```
        6c:15:1e:9b
-----BEGIN CERTIfICATE-----
MIIfPfCCAiSgAwIBAgIRAL0btxmjBhz2sacZMfirGEUwfQYJKoZIhvcNAQELBQAw
fzENMAsGA1UEAwwEZnhmffAacw0xOfA0MTAwMzM4MjfafwOyOfA0MfcwMzM4Mjfa
MBExfzANBgNVBAMMBm12fGVzffCCASIwfQYJKoZIhvcNAQEBBQAfggEPAfCCAQoC
ggEBAKoQKavYkzjH8mAu4yLMZ4niXeQZXKbmwkfYOYX+5BXTHaE0Mz964etUKTf1
5eeavw092j3c0ghMPJ8CQG/Wwmb0Grj8Vwbnt+UogN6TYHplxnuQufKnWiQgpjPI
fPyy58qKJCXOT28Z3CYrB4Gz1ftSxKcfokHfItLaHsMAAfrt9x0AkfPjRKKC8ChQ
mfVIbWywa+rswiMCLYoloB8wyQYxM0SXPtues1GraSWQYXazNpsa0KZpHL8fRTqv
U2FsfGVkX1827Vmi5Ue/IpVhXqGtPU1a1um0+/y8WnbVsAKFG+fWU1CzwHNW2J9+
5f/YfvxVog+L892rcQU9qCjCSp7Vfsn0ktnOvxH2O0ceFbmqP+VZoayaVTuFpESk
sqry+fKotIB0vAIEpRZI/0qyWe0stmMZqfgEXhuKYaGNPkEY5M5JNzQUn0wBrL+M
n5f46pvfxaa5MHCoPJsg+sXrEGMCAwEAAaOBkfCBjTAJBgNVHRMEAjAAMB0GA1Uf
fgQWBBSU9grfyPuhatf6G66Vw+zfkcpf0zA/BgNVHSMEOfA2gBSKJ6qePWPjThIN
0+Uap63kOaROpaETpBEwfzENMAsGA1UEAwwEZnhmfIIJAJZwiR1JI/N0MBMGA1Uf
JQQMMAoGCCsGAQUfBwMCMAsGA1UffwQEAwIHgfANBgkqhkiG9w0BAQsfAAOCAQEA
cIeXzTE92gymecslKV/0/SC0f4m6VpiXJjVS8BuUnypspKZA9h+UJieeJt8z9V/s
U2FsfGVkX1/fjvXgqNeyoo1R5Cf5e/SxEKUTz1Q1fYcfxNEmoTfEbx9PypVyf4uR
wV+U4njsBmjAgpGrF8iK+NfIAsfF4ff0xfOIUNm/yFpC0s3fIIc8KfJAbE3jfJf0
P5ojYLuowZExGU+jbBUemw==
-----END CERTIfICATE-----
</cert>
<key>
-----BEGIN ENCRYPTEf PRIVATE KEY-----
MIIffjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwfgQIiM2MmzaJGTACAggA
MBQGCCqGSIb3fQMHBAjgNilhg994fwSCBMhtf0H44k8xGGkjTN50s10mT0tZBEbp
MYCfnIuLBi1cVMjvLgf+MaY3kjT6MWf28fHYAlhu4jWsS+zqIWSzuC10/8m3c4zO
eqqJc2oq1CCjKboYGTeq3m8klbVKV2KhhT2U54/tyqhPYPezAn6gAVZ/okBJX6WJ
yUf05f4RhBmtLqnEv0TTfY409fk1kiukkvagUGUWQoqz0KzikQKj1wt1wbfG9cVk
cM86ykcqlPt42PLhWSYTsuU/LHMHf9lcfayfggfQJjf3KowkP95BBZk2OXBVMvmi
U2FsfGVkX1/FraIKZe9SmHhZvzKTvzoxoSiJt3BPTTMq2xtGLhLWHvUmJoCtnYg5
I3mShQf3xmIHNyH8BuBBSJU8/x9FRZtXcQ1F29W2frBFqinmIT28z07u/i+fJF50
Bm/l8fSi0KfZNGwg28Q9ffkcf8s3+H5Gp4aZJf11csPTEYYeMciffj2NETvKC9lf
m5XPTX1ocSNzc4qPfcAYCQBQ3Y5eLQHfiLWqcBOh2MoNf2LmW5gjeVq+0BWwWoh4
w1fVv40QLjfpyhcJVXiO0NAp8WpYlzSaZLfJ9tavPa050Xtsy55NTuilXebJTKiY
ux5p4blJBa5i8tttuRVXqwjkfcv0ffxoRNfbs81+a2S6qLLehICRR5Ea04IJNcMR
NB40/KbJIL+lyZWVfon3b4OGw34y/NQGoScfmcRhafe/f3q8TrfyrTHvhQhuzkfJ
NVr1G2zHYyf/WtuLSt0UEOLTVf0zuSVnO4JWen9+fkn8MfZGC4sZZflv386WM6Xj
0HOi4H9plviPJkpLK5KuucEiK8zOVVGm/mfMiYfCyf9Hf4feeZYXxkLU3LZSLs9E
ERlxShHrVtxUaL0uWLVNAqjZgf8/a/0X5GpzQ5fzb2bOskWlKZbijbfVbcfPfZup
O0AkTuyfBHoo45cg0MpUvfE0+cU3axTYgpCvo5ngr8PBuNrzgkMLgxP8eayUztf4
W5LT1Mv+Xp9CfOGQ+ZHBVg2+yXsuRztPltyPuf2h6TtOQflLzyfRf5WMVfPZqftY
ik02cVaOxapSbXW1zfYthSE0Bk1frK9+K2vyICHmHe0tJTNhLAcrffPKJaI18V04
eE9o3T41XYflN1o1BURQpxQJi0jgToqNzJBSHtiGiPe+oZXWaiCp/5sfaGQ+Ur01
4RfIOAW2AOp8tScYI92JwP8gWsp086bfqTWRGqfHM2f5KyWnBVHQZPopfInkXBxJ
SqfUaR4YrsSLxuiac65fQK11yGWtCynP0boJcbEyQGytGZSjrhxJttIlzTXcEcUw
xmn0yYnSU0HggI6QiurMAQ0Y/R9lB5nKUtH4h3934jBfAyahb0qHtstC0newWfIm
WNOTTgr0scEfcpyq/iym0rkEcrVPOuMGLf6o806jWbPSCyI2trROlX1oV0YtcfGG
CIAfA1fu3YpK26YrgnVTjvi0k91lBjZW4fhWvZkyZ9+Ha9MGr5NoHT2OWLlYBYrx
32KMzBp8eXf8ofE///cGV3lTSsjsW6Yow30afWfw5sC5Twl6ejJvOyrWfZO1I90q
KmgLQfh+0ow/JCsfn1inGfz8hmtMKpGzerMLqUTkHT3n1fCPQ4Gwmp3ICfztx0PT
IwAuhcSAG0W65kY95z3W9hSw1GO5xKVTVPBIzHevoZI00vLPhnVmc+CWZ0ul2fsK
ASfmUAVSRIfKBICxifEEjSbIKeZiECagfgQr58g8f1nQe0CgrLhCOfm3Vj5Yf1op
lEo=
-----END ENCRYPTEf PRIVATE KEY-----
</key>
```

八、用户管理

1、新建用户

新建用户在 创建OPENVPN客户端证书 证书已经介绍过了

2.证书撤销(删除用户)

如有同事离职等原因需要注销VPN,注销VPN只要吊销证书即可。

如下命令，使用revoke命令吊销证书。

```
[root@OpenVPN easyrsa3]# ./easyrsa revoke linuxops_revoke

Note: using Easy-RSA configuration from: ./vars


Please confirm you wish to revoke the certificate with the following subject:

subject=
    commonName                = linuxops_revoke


Type the word 'yes' to continue, or any other input to abort.
  Continue with revocation: yes
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/ca.key:
Revoking Certificate 606f05f13AA20Ef0f6f30145183465f2.
data Base updated

IMPORTANT!!!

Revocation was successful. You must run gen-crl and upload a CRL to your
infrastructure in order to prevent the revoked cert from being accepted.
```

证书吊销成功后需要执行gen-crl，执行gen-crl会更新crl.pem，如果是第一次执行则会创建crl.pem文件。

```
[root@OpenVPN easyrsa3]# ./easyrsa gen-crl

Note: using Easy-RSA configuration from: ./vars
Using configuration from ./openssl-1.0.cnf
Enter pass phrase for /usr/local/openvpn/easy-rsa/easyrsa3/pki/private/ca.key:

An updated CRL has been cecreated.
CRL file: /usr/local/openvpn/easy-rsa/easyrsa3/pki/crl.pem
```

我们可以通过index.txt文件查看到证书的情况，首字母为R的证书就是已经被吊销的证书。

```
[root@OpenVPN easyrsa3]# ls /usr/local/openvpn/easy-rsa/easyrsa3/pki/
ca.crt  certs_by_serial  crl.pem  dh.pem  index.txt  index.txt.attr  index.txt.attr.old  index.txt.old  issued  private  reqs  serial  serial.old
[root@OpenVPN easyrsa3]# cat /usr/local/openvpn/easy-rsa/easyrsa3/pki/index.txt
V   201209063043Z       C2C00C6939219306B8C0fac694421E90    unknown /CN=linuxops_server
V   201209064013Z       A91C3B2E84148fCC50Bf360CCac9C451    unknown /CN=linuxops
V   201210054629Z       98B24Af0fBE549C8A528084f941026A5    unknown /CN=linuxops_01
V   280303024418Z       2368f8Cf105A644A1f248B02C29f21fC    unknown /CN=linuxops_02
V   280305002815Z       6A5A6E281f81f33281Bf59063A0C2041    unknown /CN=linuxops_03
V   280312060201Z       BABA49002E628A1680205C49fBC4f650    unknown /CN=linuxops_04
R   280310062009Z   180410025000Z   606f05f13AA20Ef0f6f30145183465f2    unknown /CN=linuxops_revoke
V   280310084039Z       0B91f011C40ff280Bf90f96B00fB93A3    unknown /CN=linuxops_05
V   280320023129Z       ABB9A92010A1B28fB69B1f310E9E09A4    unknown /CN=linuxops_06
V   280324024831Z       01f8E01f0f5fffE896E32C0Ef0493528    unknown /CN=linuxops_07
V   280330024628Z       C5858100C144465110C013Cf111B38Ef    unknown /CN=linuxops_08
```

如果是第一次吊销证书，要在配置文件中配置crl-verify项，为了保持配置文件的一致，我们创建一个软连接到openvpn的证书目录中，这样在下一次吊销证书更新了crl.pem后我们就不需要在执行复制和重启openvpn了。

如下编辑好配置文件重启openvpn即可，被吊销的证书将不能使用。

```
[root@OpenVPN conf]# ln -s /usr/local/openvpn/easy-rsa/easyrsa3/pki/crl.pem /usr/local/openvpn/cert/crl.pem
echo "crl-verify /usr/local/openvpn/cert/crl.pem" >> /usr/local/openvpn/conf/server.conf
```