OpenVPN

zswu

Caveat!

The following commands, file locations is for CentOS.

If you are using FreeBSD, don’t copy-paste all below.

Why Openvpn

1.cross-platform portability

2.extensible VPN framework

3.OpenVPN uses an industrial-strength security model

TUN/TAP

TAP	TUN
Layer 2	Layer 3
behave like adapter	Less Overhead(L3)
More overhead(L2)	Only IPv4 , IPv6(Ovpn2.3)
Transfer any protocol	No Bridges!
Bridge	

Configuring Openvpn

A server/client setting can be describe as a ovpn/conf file.

At most circumstances, we will separate key/ca files to make config file clean.

server.conf

- /etc/openvpn/server/serv.conf
- cp /usr/share/doc/openvpn-2.4.6/sample/sample-config-files/server.conf /etc/openvpn/server/

A simple server config(1/2)

```
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret
dh dh2048.pem
topology subnet
server 192.168.14.0 255.255.255.0
ifconfig-pool-persist ipp.txt
client-config-dir static_clients
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
client-to-client
```

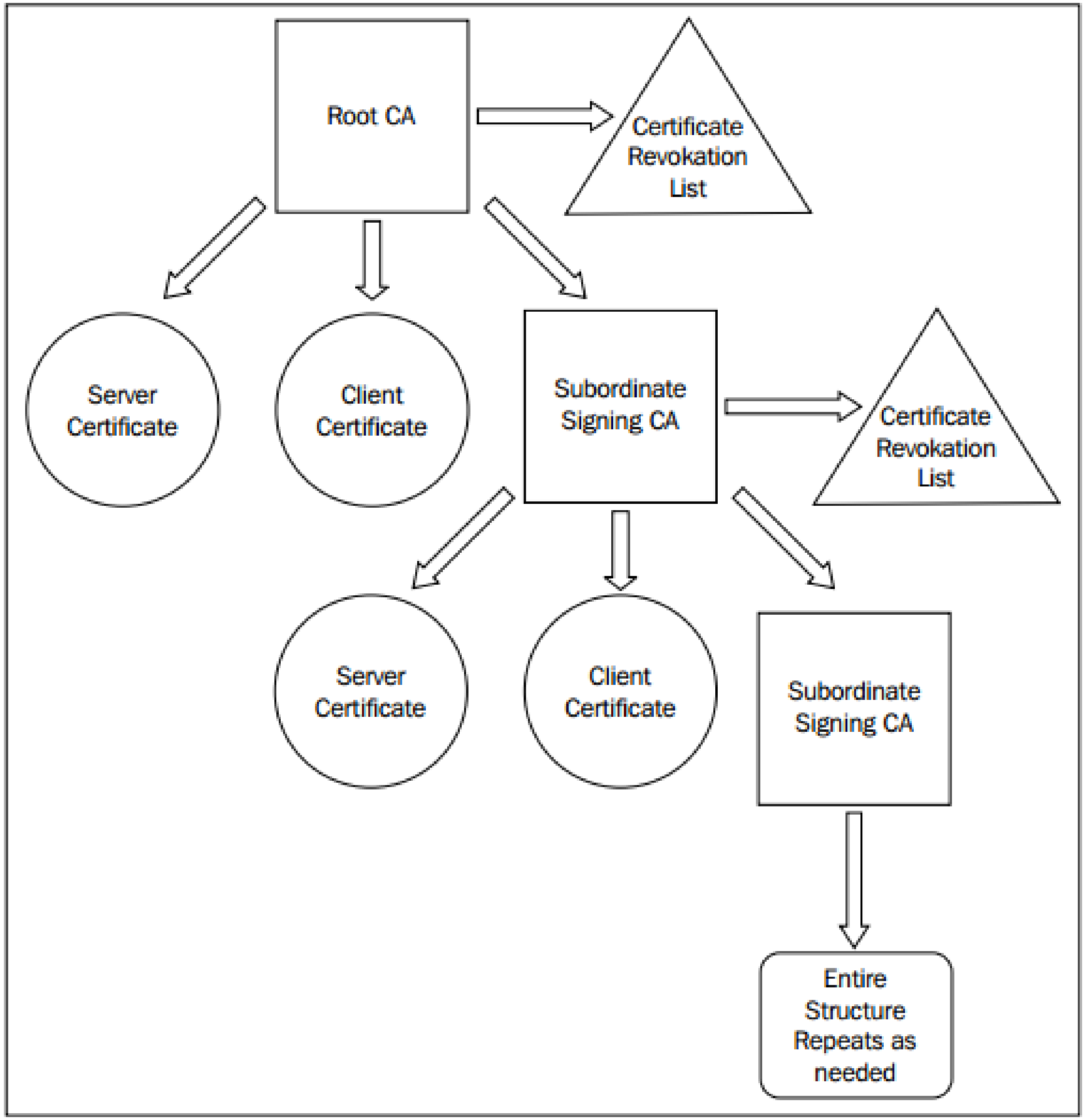
A simple server config(2/2)

```
keepalive 10 120
tls-auth ta.key 0 # This file is secret
cipher AES-256-CBC   # AES
comp-lzo
max-clients 10
user nobody
group nobody
persist-key
persist-tun
verb 5
mute 20
```

A simple client config

```
client
dev tun
proto udp
remote xxx.com 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
remote-cert-tls server
tls-auth ta.key 1
cipher AES-256-CBC
comp-lzo
verb 3
mute 20
```

X.509 PKI


Root CA
Certificate Revokation List
Server Certificate
Client Certificate
Subordinate Signing CA
Certificate Revokation List
Server Certificate
Client Certificate
Subordinate Signing CA
Entire Structure Repeats as needed

Diffie Hellman parameters

- From wikipedia:
  - Diffie–Hellman is used to secure a variety of Internet services. However, research published in October 2015 suggests that the parameters in use for many D-H Internet applications at that time are not strong enough to prevent compromise by very well-funded attackers, such as the security services of large governments.
- Generate 2048bit dhparams!

HMAC

tls-auth

The tls-auth directive adds an additional HMAC signature to all SSL/TLS handshake packets for integrity verification. Any UDP packet not bearing the correct HMAC signature can be dropped without further processing. The tls-auth HMAC signature provides an additional level of security above and beyond that provided by SSL/TLS. It can protect against:

- DoS attacks or port flooding on the OpenVPN UDP port.
- Port scanning to determine which server UDP ports are in a listening state.
- Buffer overflow vulnerabilities in the SSL/TLS implementation.
- SSL/TLS handshake initiations from unauthorized machines (while such handshakes would ultimately fail to authenticate, tls-auth can cut them off at a much earlier point).

Generate ca, cert

1.Use easy-rsa, a openvpn ca,cert generate tool

2.Do it from scratch with openssl

easy-rsa

```
# yum install easy-rsa

# mkdir /root/ca
# cd /root/ca
# /usr/share/easy-rsa/3/easyrsa init-pki
# /usr/share/easy-rsa/3/easyrsa build-ca

# cd /etc/openvpn/server
# /usr/share/easy-rsa/3/easyrsa init-pki
# /usr/share/easy-rsa/3/easyrsa gen-req [NAME] nopass
# /usr/share/easy-rsa/3/easyrsa gen-dh

# mkdir /root/client
# cd /root/client
# /usr/share/easy-rsa/3/easyrsa init-pki
# /usr/share/easy-rsa/3/easyrsa fen-req [NAME]
```

Reference:
https://community.openvpn.net/openvpn/wiki/EasyRSA3-OpenVPN-Howto
https://wiki.archlinux.org/index.php/Easy-RSA

Sign key to CA

```
# cd /root/ca
# /usr/share/easy-rsa/3/easyrsa import-req /etc/openvpn/server/pki/reqs/[NAME].req [NAME]
# /usr/share/easy-rsa/3/easyrsa import-req /root/client/pki/reqs/[NAME].req [NAME]

# /usr/share/easy-rsa/3/easyrsa sign-req server [NAME]
# /usr/share/easy-rsa/3/easyrsa sign-req client [NAME]
```

Diffie-Hellman / TLS-auth key

```
DH-KEY
# cd /etc/openvpn/server
# /usr/share/easy-rsa/3/easyrsa gen dh

AUTH KEY
# cd /etc/openvpn/server
# openvpn -genkey -secret ta.key

# cd /etc/openvpn/client
# cp ../server/ta.key ta.key
```

Package your config

Server

ca.crt

server.conf

server.key

server.crt

dh.pem

ta.key

Client

ca.crt

client.conf

client.key

client.crt

ta.key

Enable and start

```
SERVER SIDE
# cp keys,conf,crts… /etc/openvpn
# systemctl enable openvpn@CONFIG_NAME # Start at boot
 ex. systemctl enable openvpn@server
# systemctl start openvpn@CONFIG_NAME
OR
# openvpn --config ./server.conf

CLIENT SIDE
# cp keys,conf,crts… /etc/openvpn
# systemctl start openvpn@CONFIG_NAME
```

Configure NAT

```
# if you are using nftables
# add this to your table
chain postrouting {
    type nat hook postrouting priority 0;
    ip saddr 192.168.14.0/24 oifname "eth0" masquerade;
}

# if you are using iptables
# add this to your iptables.rules
 -A POSTROUTING -s 192.168.14.0/24 -o eth0 -j MASQUERADE

# if you are using firewalld
# add this to your firewall-cmd rules
firewall-cmd --zone=trusted --add-service openvpn —permanent
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -o eth0 -j MASQUERADE
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i tun0 -o eth0 -j ACCEPT # -i 是 input, -o 是
output

# sorry I don’t know how to use pf. You are on your own.
```

Confirm your vpn is working

```
# ifconifg (macOS)
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
        inet6 fe80::7a68:beac:a9c9:97cb%utun0 prefixlen 64 scopeid 0x10
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 192.168.10.2 --> 192.168.10.2 netmask 0xffffff00

# netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
0/1                192.168.10.1       UGSc         113         0   utun1
default            172.18.15.254      UGSc           1         0     en0
```

User-authentication

1. Simply by signing client certs.
2. Use Username/password

Server Side

```
Inside server.conf

# Using PAM to auth (Working with LDAP/NIS/Local Accout)
(verify-client-cert)
plugin /usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so login

# Use a shell script to auth
auth-user-pass-verify /etc/openvpn/auth.sh via-env
script-security 3 # To allow script reading passwords
```

Reference:
/usr/share/doc/openvpn-2.4.6/README.auth-pam
/etc/pam.d/login

Client Side

```
# A dialog will popup to ask you username/password
auth-user-pass
# Saving username/password into a file
auth-user-pass client.secret
# cat client.secret
Clientname
Clientpassword
```

Reference

- https://www.digitalocean.com/community/tutorials/how-to-setup-and-configure-an-openvpn-server-on-centos-7
- https://www.howtoforge.com/tutorial/how-to-install-openvpn-on-centos-7/
- https://wiki.archlinux.org/index.php/OpenVPN