Openvpn

chenshh

Caveat!

The following commands, file locations is for ArchLinux.

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

Layer 2

behave like adapter

More overhead(L2)

Transfer any protocol

Bridge

TUN

Layer 3

Less Overhead(L3)

Only IPv4 , IPv6(Ovpn2.3)

No Bridges!

Configuring Openvpn

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

At most circumstances, we will seperate

key/ca files to make config file clean.

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
```

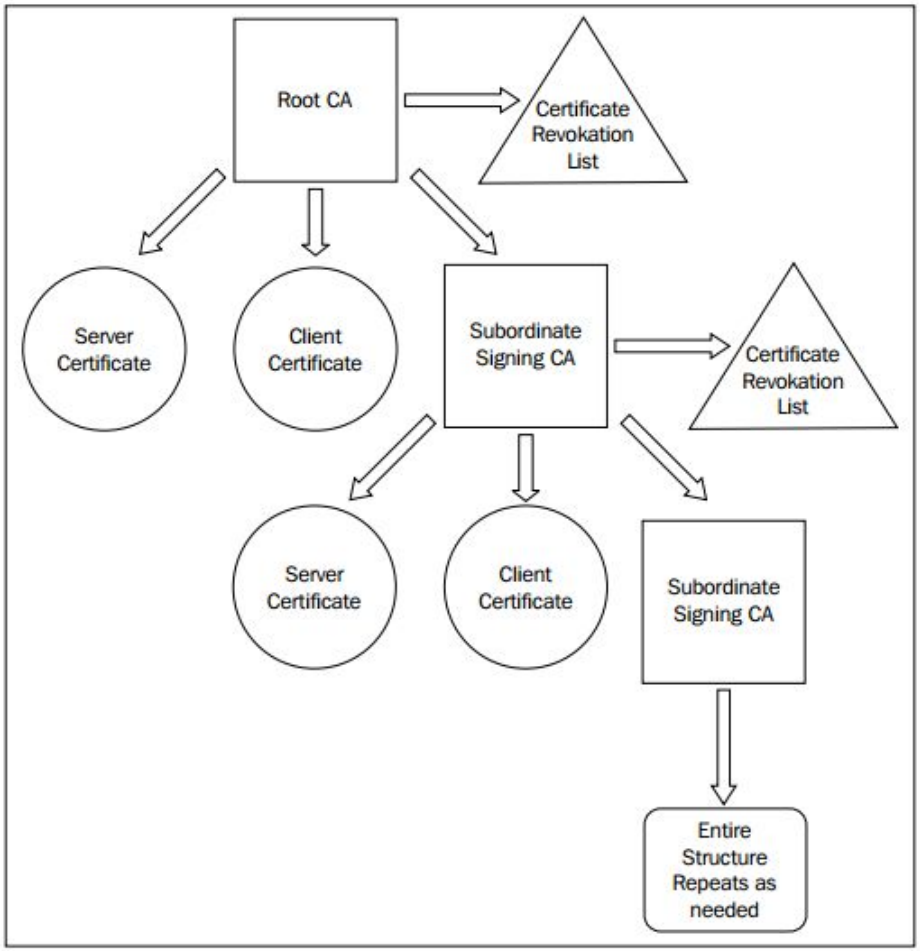
A simple server config(2/2)

```
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

```
#cp -r /usr/share/easy-rsa /root
#cd /root/easy-rsa
#$EDITOR /root/easy-rsa/vars
```

vars

```
export KEY_SIZE=2048

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Do not leave any of these fields blank.


export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="Acme Acres"
export KEY_ORG="Acme"
export KEY_EMAIL="roadrunner@acmecorp.org"
#export KEY_EMAIL=mail@host.domain
export KEY_CN=Acme-CA
export KEY_NAME=Acme-CA
export KEY_OU=""
export PKCS11_MODULE_PATH=changeme
export PKCS11_PIN=1234
```

generate!

```
# source ./vars
# ./build-ca
# ./build-key-server server
# ./build-dh
# ./build-key client
# openvpn --genkey --secret /root/easy-rsa/keys/ta.key
```

Package your config

Server

server.conf

ca.crt

server.crt

server.key

ta.key

dh2048.pem

Client

client.conf

ca.crt

client.crt

client.key

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

Confirm your vpn is working

```
# ip a
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
    default qlen 100
    link/none
    inet 192.168.14.3/24 brd 192.168.14.255 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::8c34:ce6d:d32b:2ae7/64 scope link flags 800
       valid_lft forever preferred_lft forever
# ip route
0.0.0.0/1 via 192.168.14.1 dev tun0
default via 172.17.8.2 dev wlp3s0  src 172.17.12.126  metric 302
36.234.144.197 via 172.17.8.2 dev wlp3s0
128.0.0.0/1 via 192.168.14.1 dev tun0
172.17.8.0/21 dev wlp3s0  proto kernel  scope link  src 172.17.12.126  metric 302
192.168.14.0/24 dev tun0  proto kernel  scope link  src 192.168.14.3
```

User-authentication

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

Server Side

```
# Using PAM to auth (Working with LDAP/NIS/Local Accout)
plugin /usr/lib64/openvpn/plugins/openvpn-plugin-auth-pam.so openvpn
# Use a shell script to auth
auth-user-pass-verify /etc/openvpn/auth.sh via-env
script-security 3 # To allow script reading passwords
```

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
```

Some more infos

Add these to server conf may help you

```
# Can let multiple clients connect with same cn
 duplicate-cn
# Use username as cn
 username-as-common-name
```

Per cn configs

client-config-dir directory_name
 # mkdir directory
 # $EDITOR CN_NAME