# OpenVPN

OpenVPN 是一个基于 OpenSSL 库的应用层 VPN 实现。和传统 VPN 相比，它的优点是简单易用。

OpenVPN 能在 Linux、xBSD、Mac OS X 与 Windows 2000/XP 上运行。

OpenVPN 不是一个基于 Web 的 VPN 软件，也不与 IPsec 及其他 VPN 软件包兼容。

| | | |
|---|---|---|
| LZO | lzo-2.06.tar.gz | http://www.oberhumer.com/opensource/lzo/ |
| OpenVPN Server | openvpn-2.4.6.tar.gz | https://openvpn.net |
| OpenVPN Client | openvpn-install-2.4.6-I602.exe | https://openvpn.net |
| Easy ESA | EasyRSA-3.0.4.tgz | https://github.com/OpenVPN/easy-rsa |

## 一. 软件安装

```
# 安装 pam
 yum install pam.x86_64 pam-devel.x86_64
# 安装 LZO
 tar zxvf lzo-2.06.tar.gz
 cd lzo-2.06
 ./configure
 make
 make install

# 安装 OpenVPN
 tar zxvf openvpn-2.4.6.tar.gz
 cd openvpn-2.4.6
 ./configure --prefix=/usr/local/openvpn
 make
 make install
```

## 二. 服务端证书

```
# 安装 easy-rsa
 tar zxf EasyRSA-3.0.4.tgz
 cp EasyRSA-3.0.4 /usr/local/openvpn/ -r
 cd /usr/local/openvpn/EasyRSA-3.0.4
 cp vars.example vars

# 配置 EasyRSA 环境配置文件 vim /usr/local/openvpn/EasyRSA-3.0.4/vars

# Easy-RSA 3 parameter settings
```

```
# NOTE: If you installed Easy-RSA from your distro's package manager, don't edit
# this file in place -- instead, you should copy the entire easy-rsa directory
# to another location so future upgrades don't wipe out your changes.

# HOW TO USE THIS FILE
#
# vars.example contains built-in examples to Easy-RSA settings. You MUST name
# this file 'vars' if you want it to be used as a configuration file. If you do
# not, it WILL NOT be automatically read when you call easyrsa commands.
#
# It is not necessary to use this config file unless you wish to change
# operational defaults. These defaults should be fine for many uses without the
# need to copy and edit the 'vars' file.
#
# All of the editable settings are shown commented and start with the command
# 'set_var' -- this means any set_var command that is uncommented has been
# modified by the user. If you're happy with a default, there is no need to
# define the value to its default.

# NOTES FOR WINDOWS USERS
#
# Paths for Windows  *MUST* use forward slashes, or optionally double-esscaped
# backslashes (single forward slashes are recommended.) This means your path to
# the openssl binary might look like this:
# "C:/Program Files/OpenSSL-Win32/bin/openssl.exe"

# A little housekeeping: DON'T EDIT THIS SECTION
#
# Easy-RSA 3.x doesn't source into the environment directly.
# Complain if a user tries to do this:
if [ -z "$EASYRSA_CALLER" ]; then
        echo "You appear to be sourcing an Easy-RSA 'vars' file." >&2
        echo "This is no longer necessary and is disallowed. See the section called" >&2
        echo "'How to use this file' near the top comments for more details." >&2
        return 1
fi

# DO YOUR EDITS BELOW THIS POINT

# This variable is used as the base location of configuration files needed by
# easyrsa.  More specific variables for specific files (e.g., EASYRSA_SSL_CONF)
# may override this default.
#
# The default value of this variable is the location of the easyrsa script
# itself, which is also where the configuration files are located in the
# easy-rsa tree.

#set_var EASYRSA        "${0%/*}"

# If your OpenSSL command is not in the system PATH, you will need to define the
# path to it here. Normally this means a full path to the executable, otherwise
# you could have left it undefined here and the shown default would be used.
#
# Windows users, remember to use paths with forward-slashes (or escaped
# back-slashes.) Windows users should declare the full path to the openssl
# binary here if it is not in their system PATH.

#set_var EASYRSA_OPENSSL        "openssl"
#
# This sample is in Windows syntax -- edit it for your path if not using PATH:
#set_var EASYRSA_OPENSSL        "C:/Program Files/OpenSSL-Win32/bin/openssl.exe"

# Edit this variable to point to your soon-to-be-created key directory.  By
# default, this will be "$PWD/pki" (i.e. the "pki" subdirectory of the
# directory you are currently in).
#
# WARNING: init-pki will do a rm -rf on this directory so make sure you define
```

```
# it correctly! (Interactive mode will prompt before acting.)

#set_var EASYRSA_PKI            "$PWD/pki"

# Define X509 DN mode.
# This is used to adjust what elements are included in the Subject field as the DN
# (this is the "Distinguished Name.")
# Note that in cn_only mode the Organizational fields further below aren't used.
#
# Choices are:
#   cn_only  - use just a CN value
#   org      - use the "traditional" Country/Province/City/Org/OU/email/CN format

#set_var EASYRSA_DN     "cn_only"

# Organizational fields (used with 'org' mode and ignored in 'cn_only' mode.)
# These are the default values for fields which will be placed in the
# certificate.  Don't leave any of these fields blank, although interactively
# you may omit any specific field by typing the "." symbol (not valid for
# email.)

set_var EASYRSA_REQ_COUNTRY    "CN"
set_var EASYRSA_REQ_PROVINCE   "Shanghai"
set_var EASYRSA_REQ_CITY       "Yangpu"
set_var EASYRSA_REQ_ORG        "JSVest CO.,LTD"
set_var EASYRSA_REQ_EMAIL      "ping.bao@jsvest.com"
set_var EASYRSA_REQ_OU         "Dev"

# Choose a size in bits for your keypairs. The recommended value is 2048.  Using
# 2048-bit keys is considered more than sufficient for many years into the
# future. Larger keysizes will slow down TLS negotiation and make key/DH param
# generation take much longer. Values up to 4096 should be accepted by most
# software. Only used when the crypto alg is rsa (see below.)

#set_var EASYRSA_KEY_SIZE       2048

# The default crypto mode is rsa; ec can enable elliptic curve support.
# Note that not all software supports ECC, so use care when enabling it.
# Choices for crypto alg are: (each in lower-case)
#  * rsa
#  * ec

#set_var EASYRSA_ALGO           rsa

# Define the named curve, used in ec mode only:

#set_var EASYRSA_CURVE          secp384r1

# In how many days should the root CA key expire?
set_var EASYRSA_CA_EXPIRE      365

# In how many days should certificates expire?

set_var EASYRSA_CERT_EXPIRE    365

# How many days until the next CRL publish date?  Note that the CRL can still be
# parsed after this timeframe passes. It is only used for an expected next
# publication date.

#set_var EASYRSA_CRL_DAYS       180

# Support deprecated "Netscape" extensions? (choices "yes" or "no".) The default
# is "no" to discourage use of deprecated extensions. If you require this
# feature to use with --ns-cert-type, set this to "yes" here. This support
# should be replaced with the more modern --remote-cert-tls feature.  If you do
# not use --ns-cert-type in your configs, it is safe (and recommended) to leave
# this defined to "no".  When set to "yes", server-signed certs get the
# nsCertType=server attribute, and also get any NS_COMMENT defined below in the
```

```
# nsComment field.

#set_var EASYRSA_NS_SUPPORT     "no"

# When NS_SUPPORT is set to "yes", this field is added as the nsComment field.
# Set this blank to omit it. With NS_SUPPORT set to "no" this field is ignored.

#set_var EASYRSA_NS_COMMENT     "Easy-RSA Generated Certificate"

# A temp file used to stage cert extensions during signing. The default should
# be fine for most users; however, some users might want an alternative under a
# RAM-based FS, such as /dev/shm or /tmp on some systems.

#set_var EASYRSA_TEMP_FILE      "$EASYRSA_PKI/extensions.temp"

# !!
# NOTE: ADVANCED OPTIONS BELOW THIS POINT
# PLAY WITH THEM AT YOUR OWN RISK
# !!

# Broken shell command aliases: If you have a largely broken shell that is
# missing any of these POSIX-required commands used by Easy-RSA, you will need
# to define an alias to the proper path for the command.  The symptom will be
# some form of a 'command not found' error from your shell. This means your
# shell is BROKEN, but you can hack around it here if you really need. These
# shown values are not defaults: it is up to you to know what you're doing if
# you touch these.
#
#alias awk="/alt/bin/awk"
#alias cat="/alt/bin/cat"

# X509 extensions directory:
# If you want to customize the X509 extensions used, set the directory to look
# for extensions here. Each cert type you sign must have a matching filename,
# and an optional file named 'COMMON' is included first when present. Note that
# when undefined here, default behaviour is to look in $EASYRSA_PKI first, then
# fallback to $EASYRSA for the 'x509-types' dir.  You may override this
# detection with an explicit dir here.
#
#set_var EASYRSA_EXT_DIR        "$EASYRSA/x509-types"

# OpenSSL config file:
# If you need to use a specific openssl config file, you can reference it here.
# Normally this file is auto-detected from a file named openssl-easyrsa.cnf from the
# EASYRSA_PKI or EASYRSA dir (in that order.) NOTE that this file is Easy-RSA
# specific and you cannot just use a standard config file, so this is an
# advanced feature.

#set_var EASYRSA_SSL_CONF       "$EASYRSA/openssl-easyrsa.cnf"

# Default CN:
# This is best left alone. Interactively you will set this manually, and BATCH
# callers are expected to set this themselves.

#set_var EASYRSA_REQ_CN         "ChangeMe"

# Cryptographic digest to use.
# Do not change this default unless you understand the security implications.
# Valid choices include: md5, sha1, sha256, sha224, sha384, sha512

#set_var EASYRSA_DIGEST         "sha256"

# Batch mode. Leave this disabled unless you intend to call Easy-RSA explicitly
# in batch mode without any user input, confirmation on dangerous operations,
# or most output. Setting this to any non-blank string enables batch mode.

#set_var EASYRSA_BATCH          ""
```

```
-- 生成 PKI
./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/EasyRSA-3.0.4/pki

-- 创建 CA
./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
............+++
.................................................+++
writing new private key to '/usr/local/openvpn/EasyRSA-
3.0.4/pki/private/ca.key.jln95UchZF'
-----
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 called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]: [回车]

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/usr/local/openvpn/EasyRSA-3.0.4/pki/ca.crt

-- 创建证书
./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/EasyRSA-
3.0.4/pki/private/server.key.1i2MwYE2aC'
-----
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 called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]: [回车]

Keypair and certificate request completed. Your files are:
req: /usr/local/openvpn/EasyRSA-3.0.4/pki/reqs/server.req
key: /usr/local/openvpn/EasyRSA-3.0.4/pki/private/server.key

-- 签约证书
./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                = server

Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from ./openssl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :PRINTABLE:'server'
Certificate is to be certified until Sep  8 08:43:19 2028 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /usr/local/openvpn/EasyRSA-3.0.4/pki/issued/server.crt


-- 创建 Diffie-Hellman
./easyrsa gen-dh

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 created at /usr/local/openvpn/EasyRSA-3.0.4/pki/dh.pem

# 处理服务端所需要的证书

cd /usr/local/openvpn
mkdir /usr/local/openvpn/certs


cp EasyRSA-3.0.4/pki/dh.pem /usr/local/openvpn/certs/
cp EasyRSA-3.0.4/pki/ca.crt /usr/local/openvpn/certs/
cp EasyRSA-3.0.4/pki/issued/server.crt /usr/local/openvpn/certs/
cp EasyRSA-3.0.4/pki/private/server.key /usr/local/openvpn/certs/
```

## 三. 客户端证书

```
# 开始为创建客户端证书，证书名为 client1

cd /usr/local/openvpn/EasyRSA-3.0.4
./easyrsa gen-req client1 nopass
Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
..............+++
.........................+++
writing new private key to '/usr/local/openvpn/EasyRSA-
3.0.4/pki/private/client1.key.BEJdekqtms'
-----
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 called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [client1]: [回车]

Keypair and certificate request completed. Your files are:
req: /usr/local/openvpn/client/pki/reqs/client1.req
key: /usr/local/openvpn/client/pki/private/client1.key

/* 备注 */
/* 导入别人提供的 req 文件方法如下： */
/* ./easyrsa import-req /usr/local/openvpn/EasyRSA-3.0.4/pki/reqs/client1.req client1 */

-- 对 client1 进行签名
./easyrsa sign client client1

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                = jsw


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from ./openssl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :PRINTABLE:'jsw'
Certificate is to be certified until Sep  8 09:04:00 2028 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /usr/local/openvpn/EasyRSA-3.0.4/pki/issued/client1.crt

# 客户端所需 3 个证书文件下载至 Windows 端

/usr/local/openvpn/EasyRSA-3.0.4/pki/ca.crt
/usr/local/openvpn/EasyRSA-3.0.4/pki/issued/client1.crt
/usr/local/openvpn/EasyRSA-3.0.4/pki/private/client1.key
```

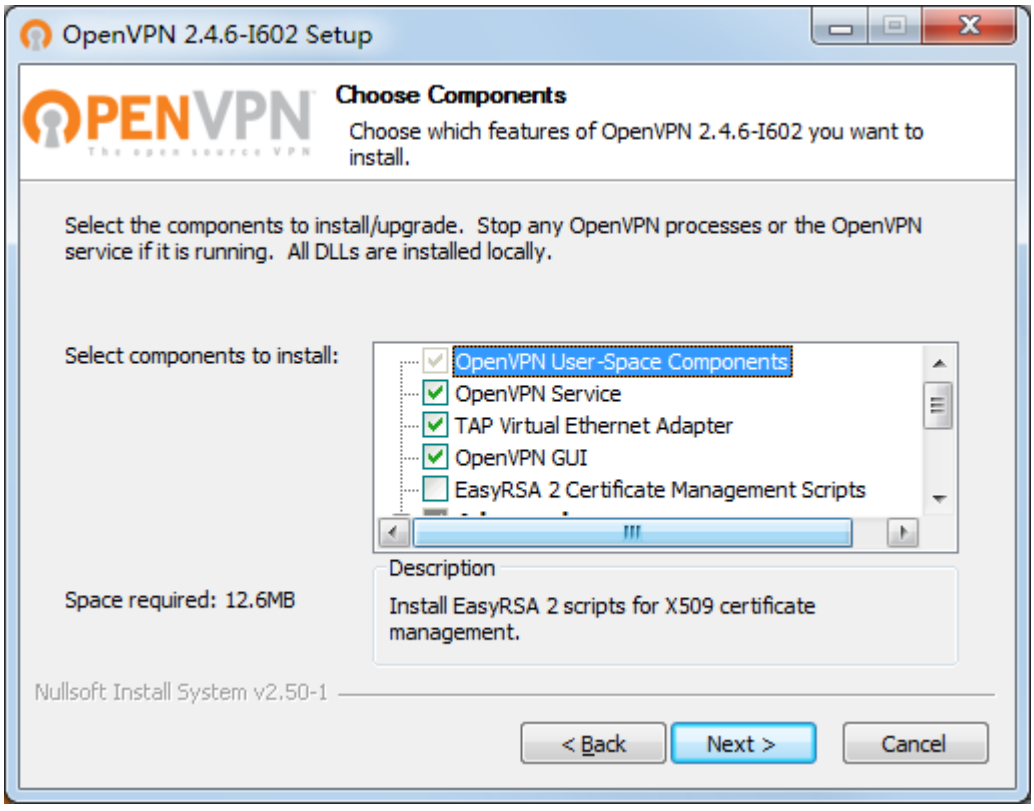
## 四. 服务端配置（本实例使用证书认证）

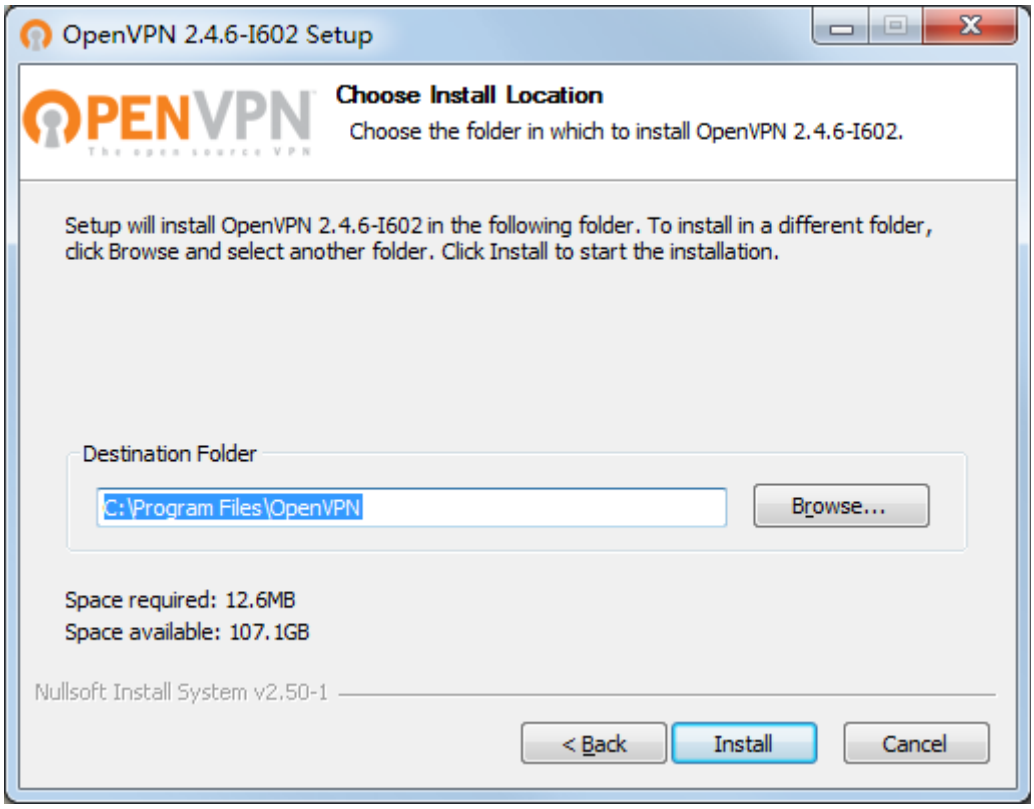
```
# 配置 openvpn 服务端 （tun 点对点 tap 以太网，client-to-client 客户端互联，tap 测试成功）
touch /usr/local/openvpn/server.conf
vim /usr/local/openvpn/server.conf

port 1194
proto tcp
dev tap
ca /usr/local/openvpn/certs/ca.crt
```

```
cert /usr/local/openvpn/certs/server.crt
key /usr/local/openvpn/certs/server.key
dh /usr/local/openvpn/certs/dh.pem
server 172.16.6.0 255.255.255.0                          -- 客户端分配的地址池
push "redirect-gateway"
# push "route 192.168.1.0 255.255.255.0 192.168.2.1"    -- 客户端路由，无网关则到 vpn server
push "dhcp-option DNS 172.16.3.21"
push "dhcp-option DNS 1.1.1.1"
keepalive 10 120
comp-lzo
persist-key
persist-tun
client-to-client
duplicate-cn

# 防火墙及转发设置
 # echo 1 > /proc/sys/net/ipv4/ip_forward
 sysctl -e net.ipv4.ip_forward=1
 iptables -t nat -A POSTROUTING -s 172.16.6.0/24 -j MASQUERADE

# 启动 OpenVPN
 /usr/local/openvpn/sbin/openvpn --daemon --config /usr/local/openvpn/server.conf
```

## 五. 客户端安装

1. 安装 OpenVPN 客户端 openvpn-install-2.4.6-I602.exe







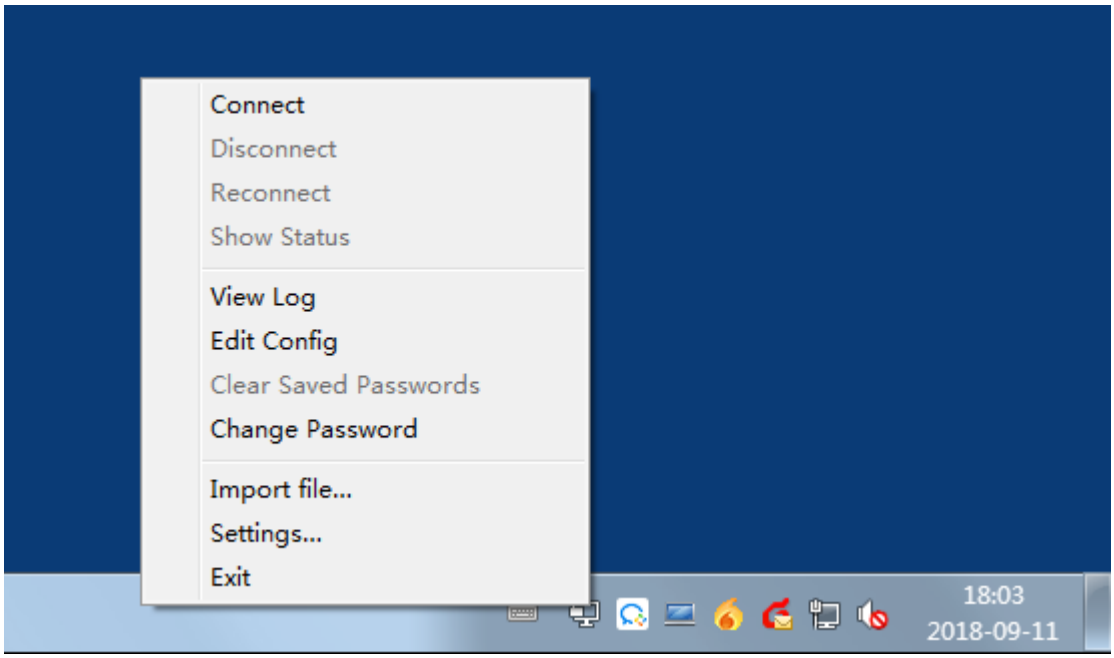
3. 配置客户端

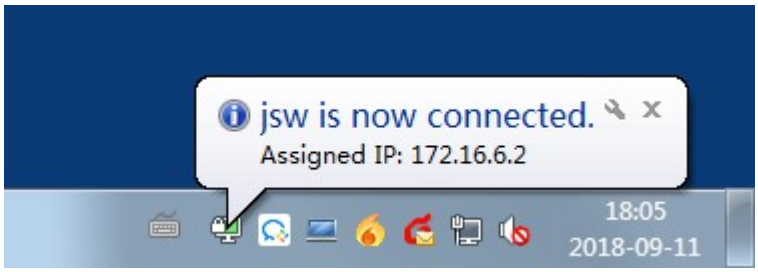
```
# 复制下载的 ca.crt client1.crt client1.key 三个文件至 C:\Program Files\OpenVPN\config
copy ca.crt client1.crt client1.key "C:\Program Files\OpenVPN\config\172.16.3.21\"
```

```
# 通过配置文件 C:\Program Files\OpenVPN\config\172.16.3.21\client1.ovpn 指定服务端，内容如下
client
dev tap
proto tcp
remote 172.16.3.21 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
remote-cert-tls server
comp-lzo
verb 4
```

3. 使用客户端

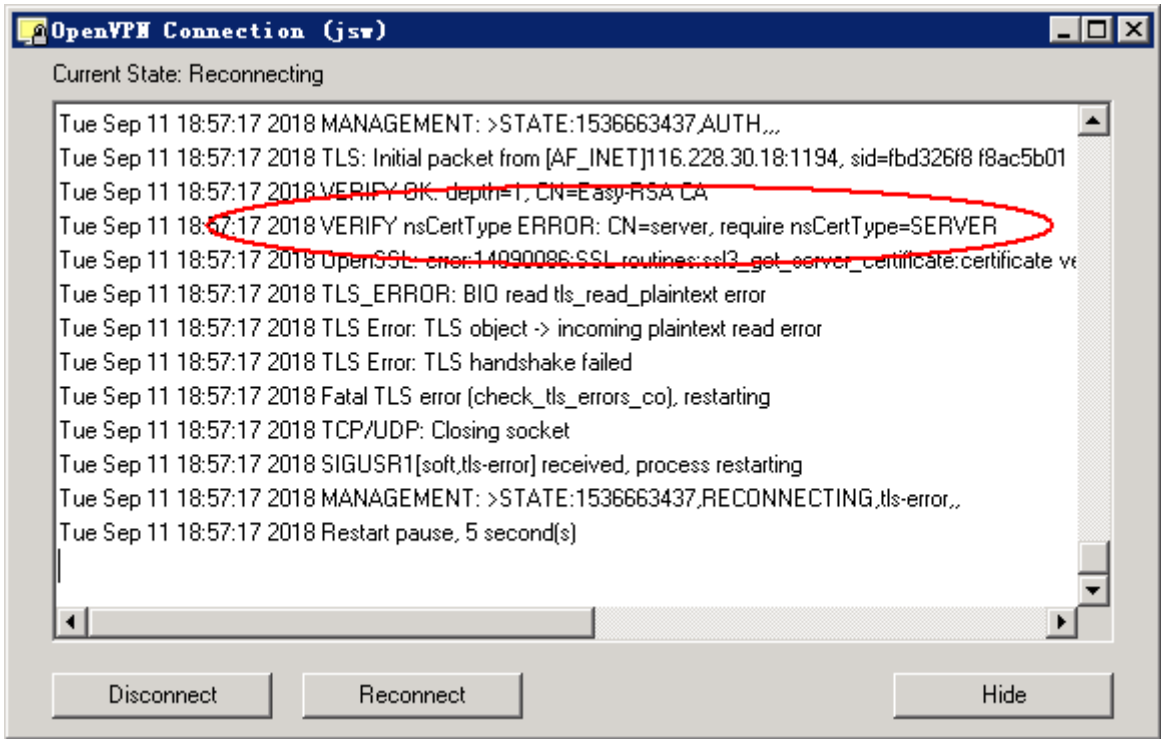
执行图标 OpenVPN GUI 或运行 "C:\Program Files\OpenVPN\bin\openvpn-gui.exe"
右键点击状态栏图标，选择 connect



## 六. 常见问题

1. 客户端报错




解决办法： 注释客户端配置文件内 `#ns-cert-type server`