老男孩教育-隧道服务-OpenVPN

☑ vpn概述

☑ **OpenVPN应用场景** 🌟 🌟🌟🌟🌟

☑ OpenVPN原理(网络)

☑ **OpenVPN服务端** 🌟

☑ **OpenVPN客户端** 🌟

☑ 连接与使用(客户连接服务端)

☑ **OpenVPN访问服务器内网** 🌟🌟🌟🌟

☑ **OpenVPN加密访问(访问的时候输入用户名密码)** 🌟🌟🌟🌟🌟

1. vpn概述

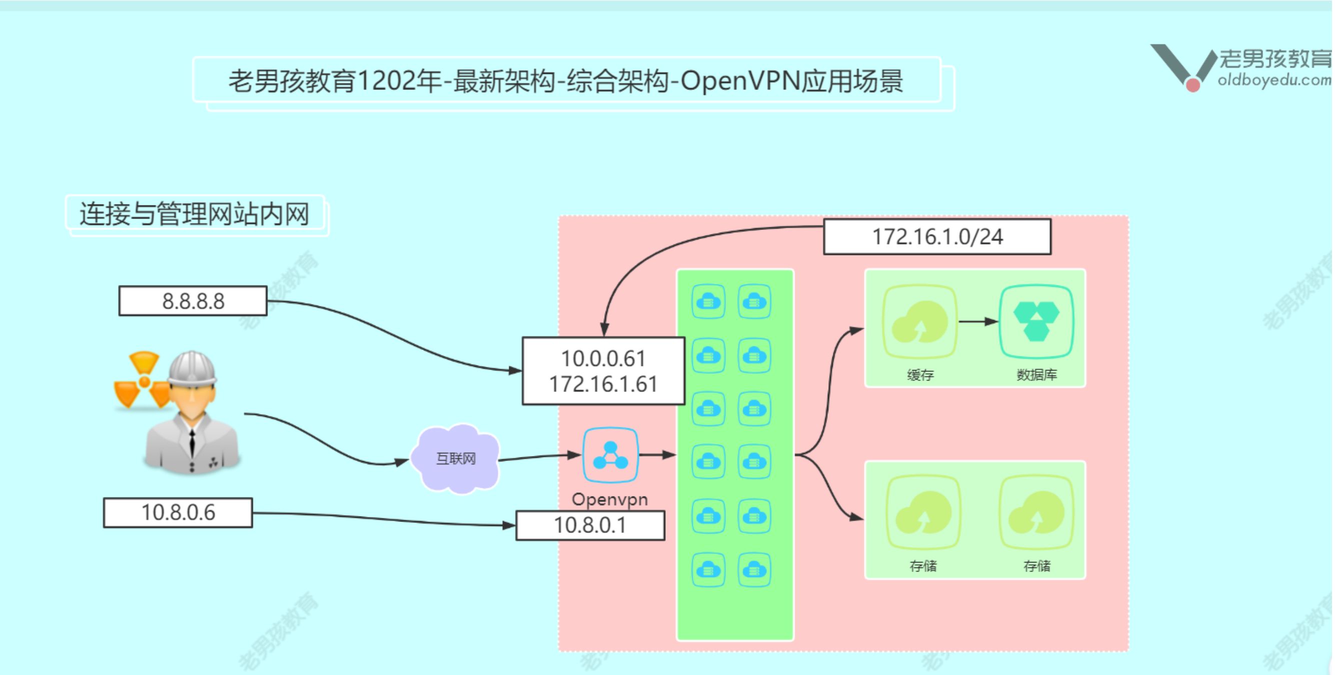
- 两点如何传输数据最安全
    - 方案1: 专线
    - 方案2: 硬件设备3层路由器 , 硬件vpn设备 vpn virtual private network 虚拟专有网络
    - 方案3: **开源软件**
        - pptp 使用最简单,不是很稳定,依赖于硬件设备的支持.
        - **OpenVPN** 实现用户/运维/开发,访问网站内网.
        - IpSEC
        - OpenSwan

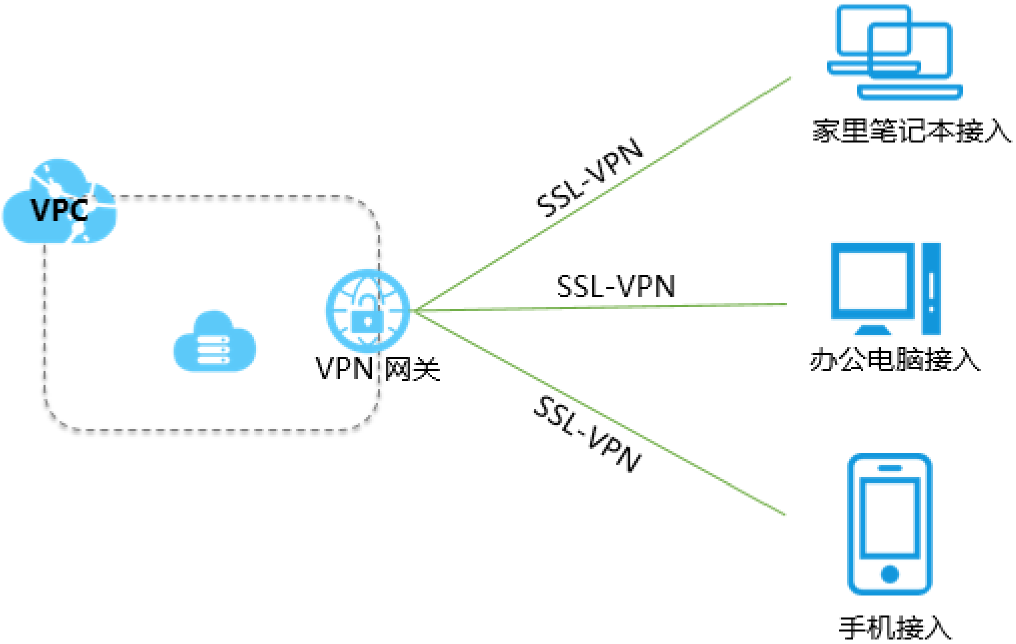
2.OpenVPN应用场景

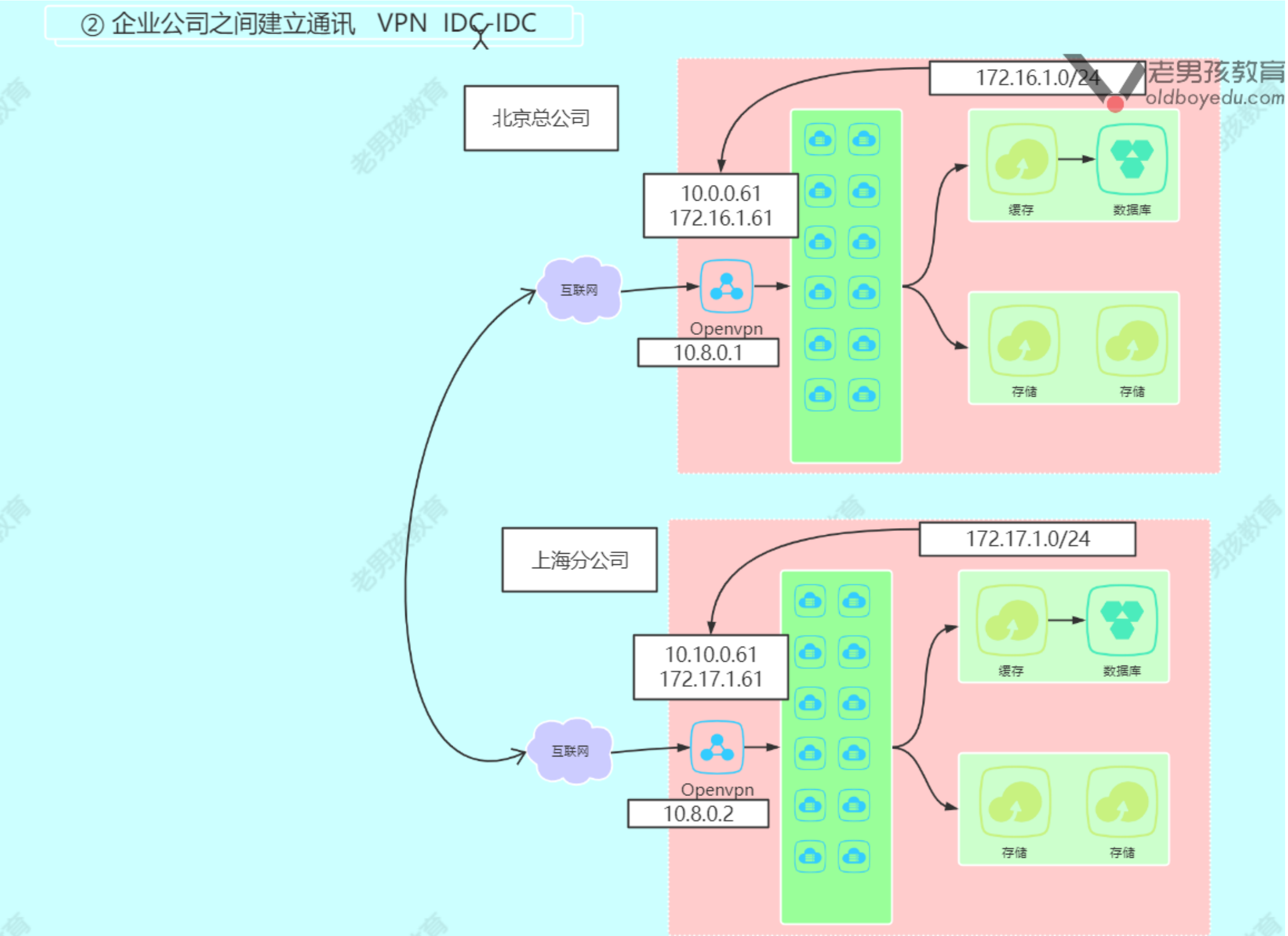
主机远程访问服务器设备 VPN 访问单台设备 🌟🌟🌟🌟🌟

个人出差访问服务设备 VPN

企业公司之间建立通讯 VPN IDC-IDC


老男孩教育1202年-最新架构-综合架构-OpenVPN应用场景
老男孩教育
oldboyedu.com
连接与管理网站内网
8.8.8.8
10.0.0.61
172.16.1.61
172.16.1.0/24
互联网
Openvpn
10.8.0.6
10.8.0.1
缓存
数据库
存储
存储


VPC
VPN 网关
SSL-VPN
SSL-VPN
SSL-VPN
家里笔记本接入
办公电脑接入
手机接入



3. OpenVPN原理(网络)

4. OpenVPN服务端配置

☑ 创建CA证书

☑ server密钥

☑ client密钥

□ OpenVPN服务端配置文件

```
# easy-rsa
yum install -y  esay-rsa
[root@m01 ~]# rpm -ql easy-rsa

/usr/share/doc/easy-rsa-3.0.8/vars.example




/usr/share/easy-rsa/3.0.8
/usr/share/easy-rsa/3.0.8/easyrsa
/usr/share/easy-rsa/3.0.8/openssl-easyrsa.cnf
/usr/share/easy-rsa/3.0.8/x509-types
/usr/share/easy-rsa/3.0.8/x509-types/COMMON
/usr/share/easy-rsa/3.0.8/x509-types/ca
/usr/share/easy-rsa/3.0.8/x509-types/client
/usr/share/easy-rsa/3.0.8/x509-types/code-signing
/usr/share/easy-rsa/3.0.8/x509-types/email
/usr/share/easy-rsa/3.0.8/x509-types/kdc
/usr/share/easy-rsa/3.0.8/x509-types/server
```

```
/usr/share/easy-rsa/3.0.8/x509-types/serverClient
/usr/share/licenses/easy-rsa-3.0.8
/usr/share/licenses/easy-rsa-3.0.8/gpl-2.0.txt



#1. 下载生成证书的文件
easy-rsa

#2. 准备vars,充当CA权威机构:
 mkdir /opt/easy-rsa
 cd /opt/easy-rsa/
/usr/bin/cp -a /usr/share/easy-rsa/3.0.8/* ./
/usr/bin/cp -a /usr/share/doc/easy-rsa-3.0.8/vars.example ./vars
[root@vpn easy-rsa]# > vars
[root@m01 easy-rsa]# cat vars
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
set_var EASYRSA_DN  "cn_only"
set_var EASYRSA_REQ_COUNTRY "CN"
set_var EASYRSA_REQ_PROVINCE "Beijing"
set_var EASYRSA_REQ_CITY "Shanghai"
set_var EASYRSA_REQ_ORG "oldboy"
set_var EASYRSA_REQ_EMAIL "oldboy@qq.comm"
set_var EASYRSA_NS_SUPPORT "yes"

[root@m01 /opt/easy-rsa]# tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── vars                    #ca证书信息
└── x509-types
    ├── ca
    ├── client
    ├── code-signing
    ├── COMMON
    ├── email
    ├── kdc
    ├── server
    └── serverClient

1 directory, 11 files

##1.初始化，在当前目录创建PKI目录，用于存储证书
[root@m01 easy-rsa]# ./easyrsa init-pki

[root@m01 /opt/easy-rsa]# ./easyrsa init-pki

Note: using Easy-RSA configuration from: /opt/easy-rsa/vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /opt/easy-rsa/pki
注意：使用Easy RSA配置来自：/opt/Easy RSA/vars
初始化pki完成；您现在可以创建一个CA或多个请求。
新创建的PKI目录是：/opt/easy rsa/PKI

[root@m01 /opt/easy-rsa]# tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── pki
│   ├── openssl-easyrsa.cnf
│   ├── private
│   ├── reqs
│   └── safessl-easyrsa.cnf
├── vars
```

```
└── x509-types
    ├── ca
    ├── client
    ├── code-signing
    ├── COMMON
    ├── email
    ├── kdc
    ├── server
    └── serverClient

4 directories, 13 files
###2.创建根证书，会提示设置密码，用于ca对之后生成的server和client证书签名时使用，其他可默认
##温馨提示: 加上密码
[root@m01 easy-rsa]# ./easyrsa build-ca
root@m01 /opt/easy-rsa]# ./easyrsa build-ca

Note: using Easy-RSA configuration from: /opt/easy-rsa/vars
Using SSL: openssl OpenSSL 1.0.2k-fips  26 Jan 2017

Enter New CA Key Passphrase:           #设置个密码
Re-Enter New CA Key Passphrase:        #确认密码
Generating RSA private key, 2048 bit long modulus
......+++
.............................................+++
e is 65537 (0x10001)
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:
/opt/easy-rsa/pki/ca.crt    #证书名字及目录


###3.创建server端证书和私钥文件，nopass表示不加密私钥文件，其他可默认
[root@m01 easy-rsa]# ./easyrsa gen-req server nopass

Keypair and certificate request completed. Your files are:
req: /opt/easy-rsa/pki/reqs/server.req   #证书请求文件
key: /opt/easy-rsa/pki/private/server.key #私钥


#4.给server端证书签名，首先是对一些信息的确认，可以输入yes，然后创建ca根证书时设置的密码
[root@m01 easy-rsa]# ./easyrsa sign server server

Certificate created at: /opt/easy-rsa/pki/issued/server.crt   #证书文件
key: /opt/easy-rsa/pki/private/server.key #私钥


[root@m01 /opt/easy-rsa]# tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── pki
│   ├── ca.crt
│   ├── certs_by_serial
│   │   └── 227D37289FF5862F276462B0A00419C3.pem
│   ├── index.txt
│   ├── index.txt.attr
│   ├── index.txt.attr.old
│   ├── index.txt.old
│   ├── issued
│   │   └── server.crt
│   ├── openssl-easyrsa.cnf
│   ├── private
│   │   ├── ca.key
│   │   └── server.key
│   ├── renewed
│   │   ├── certs_by_serial
│   │   ├── private_by_serial
│   │   └── reqs_by_serial
│   ├── reqs
│   │   └── server.req
│   ├── revoked
│   │   ├── certs_by_serial
│   │   ├── private_by_serial
│   │   └── reqs_by_serial
│   ├── safessl-easyrsa.cnf
│   ├── serial
│   └── serial.old
├── vars
└── x509-types
    ├── ca
    ├── client
    ├── code-signing
    ├── COMMON
    ├── email
    ├── kdc
    ├── server
    └── serverClient

14 directories, 25 files





#5.创建Diffie-Hellman文件，秘钥交换时的Diffie-Hellman算法
[root@m01 easy-rsa]# ./easyrsa gen-dh




#服务端的 ca证书   服务端证书(公钥)和私钥
├── pki
│   ├── ca.crt   #ca证书
│   ├── private
│   │   └── server.key   #服务端证书(公钥)
│   ├── issued
│   │   └── server.crt   #服务端私钥
│   ├── dh.pem           #认证算法


#6.创建client端证书和私钥文件，nopass表示不加密私钥文件，其他可默认
[root@m01 easy-rsa]# ./easyrsa gen-req client nopass
Keypair and certificate request completed. Your files are:
req: /opt/easy-rsa/pki/reqs/client.req
key: /opt/easy-rsa/pki/private/client.key

#7.给client端证书签名，首先是对一些信息的确认，可以输入yes，然后创建ca根证书时设置的密码
[root@m01 easy-rsa]# ./easyrsa sign client client
req: /opt/easy-rsa/pki/reqs/client.crt
key: /opt/easy-rsa/pki/private/client.key


│   ├── issued
│   │   ├── client.crt
│   ├── private
│   │   ├── client.key


#汇总
目前为止的目录结构及主要内容
[root@m01 /opt/easy-rsa]# tree
.
```

```
├── easyrsa   #管理命令
├── pki
|   ├── ca.crt  #ca证书  服务端与客户端都是用
|   ├── dh.pem  #认证算法 服务端
|   ├── issued
|   |   ├── client.crt   #客户端证书
|   |   └── server.crt   #服务端证书
|   ├── private
|   |   ├── ca.key
|   |   ├── client.key   #客户端私钥
|   |   └── server.key   #服务端私钥
```

- 安装openvpn

```
#服务端配置文件
[root@web01 openvpn]# vim /etc/openvpn/server.conf
port 1194                               #端口
proto udp                               #协议
dev tun                                 #采用路由隧道模式tun
ca ca.crt                               #ca证书文件位置   /etc/openvpn   /etc/opnevpn/server
                                        server/ca.crt
cert server/server.crt                         #服务端公钥名称
                                                #/etc/openvpn/server/
key server/server.key                          #服务端私钥名称
                                                #/etc/openvpn/server/
dh dh.pem                               #交换证书 校验算法 /etc/openvpn

server 10.8.0.0 255.255.255.0           #给客户端分配地址池，
                                        #注意：不能和VPN服务器内网网段有相同
push "route 172.16.1.0 255.255.255.0"   #允许客户端访问内网172.16.1.0网段
push "route 172.16.2.0 255.255.255.0"

#ifconfig-pool-persist ipp.txt           #地址池记录文件位置 未来让openvpn 客户端固定ip地址使用的.
keepalive 10 120                        #存活时间，10秒ping一次,120 如未收到响应则视为断线
max-clients 100                         #最多允许100个客户端连接
status openvpn-status.log               #日志记录位置  openvpn状态
log /var/log/openvpn.log                #openvpn日志记录位置
verb 3                                  #openvpn版本
client-to-client                        #客户端与客户端之间支持通信
persist-key     #通过keepalive检测超时后，重新启动VPN，不重新读取keys，保留第一次使用的keys。
persist-tun     #检测超时后，重新启动VPN，一直保持tun是linkup的。否则网络会先linkdown然后再linkup
duplicate-cn    #客户端密钥(证书和私钥)是否可以重复

#复制证书及密钥

 cp  /opt/easy-rsa/pki/ca.crt /etc/openvpn/
 cp /opt/easy-rsa/pki/issued/server.crt   /opt/easy-rsa/pki/private/server.key   /etc/openvpn/
 cp /opt/easy-rsa/pki/dh.pem   /etc/openvpn/

#启动
systemctl start openvpn@server
systemctl enable openvpn@server

#检查进程与端口
[root@m01 ~]# ip  a  s  tun0
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::a012:99d:88da:14dd/64 scope link flags 800
       valid_lft forever preferred_lft forever
[root@m01 ~]#
[root@m01 ~]# ss -lntup |grep 1194
udp    UNCONN     0      0          *:1194                  *:*                   users:(("openvpn",pid=12564,fd=6))
[root@m01 ~]# ps -ef |grep openvpn
root      12564      1  0 17:44 ?        00:00:00 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf
```

```


#服务端日志:
ROUTE_GATEWAY 10.0.0.2/255.255.255.0 IFACE=eth0 HWADDR=00:xxxxxxx  #openvpn发下当前系统网关
TUN/TAP device tun0 opened   #添加openvpn虚拟网卡 tun0
TUN/TAP TX queue length set to 100
/sbin/ip link set dev tun0 up mtu 1500
/sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2  #openvpn 给tun0设置ip 10.8.0.1
/sbin/ip route add 10.8.0.0/24 via 10.8.0.2              #在系统中添加路由信息

route -n
10.8.0.0        10.8.0.2        255.255.255.0   UG    0      0        0 tun0





#客户端配置文件


```

5. OpenVPN客户端

- windows client.ovpn
- linux   /etc/openvpn/client/client.conf
- xxxx linux/unix

5.1 windows客户端

```
#windows

C:\Program Files\OpenVPN\config
oldboyedu.com #目录
    ca.crt
    client.crt
    client.key
    client.ovpn  #client.conf
lidaoav.com   #目录
    ca.crt
    client.crt
    client.key
    client.ovpn  #client.conf


#client.ovpn
root@openvpn-client ~]# cat /etc/openvpn/clinet.ovpn
client                  #指定当前VPN是客户端
dev tun                 #使用tun隧道传输协议
proto udp               #使用udp协议传输数据
remote 10.0.0.61 1194   #openvpn服务器IP地址端口号
resolv-retry infinite   #断线自动重新连接，在网络不稳定的情况下非常有用
nobind                  #不绑定本地特定的端口号
ca ca.crt               #指定CA证书的文件路径
cert client.crt         #指定当前客户端的证书文件路径
key client.key          #指定当前客户端的私钥文件路径
verb 3                  #指定日志文件的记录详细级别，可选0-9，等级越高日志内容越详细
persist-key             #通过keepalive检测超时后，重新启动VPN，不重新读取keys，保留第一次使用的keys
persis

#客户端日志
MANAGEMENT: >STATE:1622455496,ASSIGN_IP,,10.8.0.6,,,,  #客户端ip地址

C:\WINDOWS\system32\route.exe ADD 172.16.1.0 MASK 255.255.255.0 10.8.0.5  #客户端想要访问 172.16.1.0/24 网段请走 10.8.0.5
Route addition via service succeeded
```

```
C:\WINDOWS\system32\route.exe ADD 10.8.0.0 MASK 255.255.255.0 10.8.0.5





#实现 客户访问网站的内网
route add -net 10.8.0.0/24 gw 172.16.1.61



tcpdump -i eth1   -nn    icmp   #
IP 10.8.0.6 > 172.16.1.51: ICMP echo request, id 1, seq 1094, length 40 #   请求
IP 172.16.1.51 > 10.8.0.6: ICMP echo reply, id 1, seq 1094, length 40   # 响应


课下测试  tcpdump抓取 http请求????
```

5.2 opevpn linux客户端

```
#client.ovpn
root@openvpn-client ~]# cat /etc/openvpn/client/clinet.conf
client                  #指定当前VPN是客户端
dev tun                 #使用tun隧道传输协议
proto udp               #使用udp协议传输数据
remote 10.0.0.61 1194   #openvpn服务器IP地址端口号
resolv-retry infinite   #断线自动重新连接，在网络不稳定的情况下非常有用
nobind                  #不绑定本地特定的端口号
ca client/ca.crt               #指定CA证书的文件路径
cert client/client.crt         #指定当前客户端的证书文件路径
key client/client.key          #指定当前客户端的私钥文件路径
verb 3                  #指定日志文件的记录详细级别，可选0-9，等级越高日志内容越详细
persist-k

/usr/sbin/openvpn --cd /etc/openvpn/ --config client/client.conf


[root@m01 ~]# systemctl cat openvpn@client.service
# /usr/lib/systemd/system/openvpn@.service
[Unit]
Description=OpenVPN Robust And Highly Flexible Tunneling Application On %I
After=network.target

[Service]
Type=notify
PrivateTmp=true
ExecStart=/usr/sbin/openvpn --cd /etc/openvpn/ --config client/%i.conf

[Install]
WantedBy=multi-user.target


[root@openvpn-client ~]# openvpn --daemon --cd /etc/openvpn --config client/client.conf --log-append /var/log/openvpn-client.log

# --daemon：openvpn以daemon方式启动。
# --cd dir：配置文件的目录，openvpn初始化前，先切换到此目录。
# --config file：客户端配置文件的路径。
# --log-append file：日志文件路径，如果文件不存在会自动创建。
```

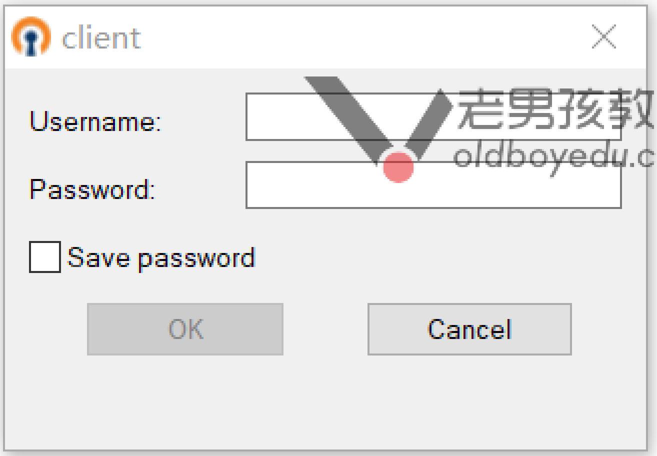
6. OpenVPN加密/认证

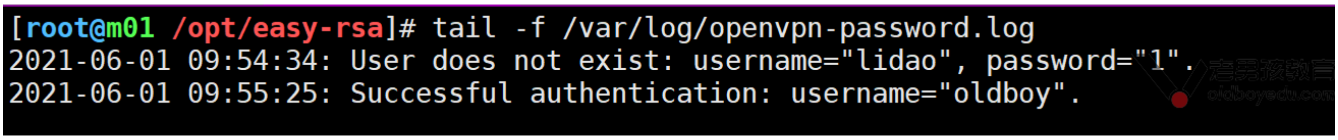
官方建议		
ca.crt	openvpn服务端	
server.crt	openvpn服务端	
server.key (dh.pem)	openvpn服务端	
ca.crt	openvpn 客户端01	
client1.crt	openvpn 客户端01	
client1.key	openvpn 客户端01	
ca.crt	openvpn 客户端02	
client2.crt	openvpn 客户端02	
client2.key	openvpn 客户端02	

最佳实践		
ca.crt	openvpn服务端	
server.crt	openvpn服务端	
server.key (dh.pem)	openvpn服务端	
ca.crt	openvpn 客户端01	
client.crt	openvpn 客户端01	
client.key	openvpn 客户端01	
	登录的时候输入用户和密码 oldboy 123456	
ca.crt	openvpn 客户端02	
client.crt	openvpn 客户端02	
client.key	openvpn 客户端02	
	登录的时候输入用户和密码 lidao 123456	

```
#openvpn  server
1.先配置服务端支持密码认证:
[root@web01 ~]# vim /etc/openvpn/server.conf
script-security 3                                 #允许使用自定义脚本
auth-user-pass-verify /etc/openvpn/check.sh via-env #指定认证脚本
```

```
username-as-common-name                                    #用户密码登陆方式验证



2.编写/etc/openvpn/check.sh 脚本文件
[root@m01 ~]# cat /etc/openvpn/check.sh
#!/bin/sh
#desc: openvpn  uesr check   scripts
#author: by  oldboylinux
############################################################
PASSFILE="/etc/openvpn/openvpnfile"             #密码文件 用户名 密码明文
LOG_FILE="/var/log/openvpn-password.log"        #用户登录情况的日志
TIME_STAMP=`date "+%Y-%m-%d %T"`

    if [ ! -r "${PASSFILE}" ]; then
      echo "${TIME_STAMP}: Could not open password file \"${PASSFILE}\" for reading." >> ${LOG_FILE}
      exit 1
    fi

    CORRECT_PASSWORD=`awk '!/^;/&&!/^#/&&$1=="'${username}'"{print $2;exit}' ${PASSFILE}`

    if [ "${CORRECT_PASSWORD}" = "" ]; then
      echo "${TIME_STAMP}: User does not exist: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
          exit 1
    fi
    if [ "${password}" = "${CORRECT_PASSWORD}" ]; then
      echo "${TIME_STAMP}: Successful authentication: username=\"${username}\"." >> ${LOG_FILE}
      exit 0
    fi
    echo "${TIME_STAMP}: Incorrect password: username=\"${username}\", password=\"${password}\"." >> ${LOG_FILE}
exit 1

3. 设置权限
chmod 700  /etc/openvpn/check.sh

4. 创建用户
cat > /etc/openvpn/openvpnfile<<EOF
oldboy 1
lidao:1
EOF

5. 重启服务端



#openvpn 客户端
auth-user-pass

```



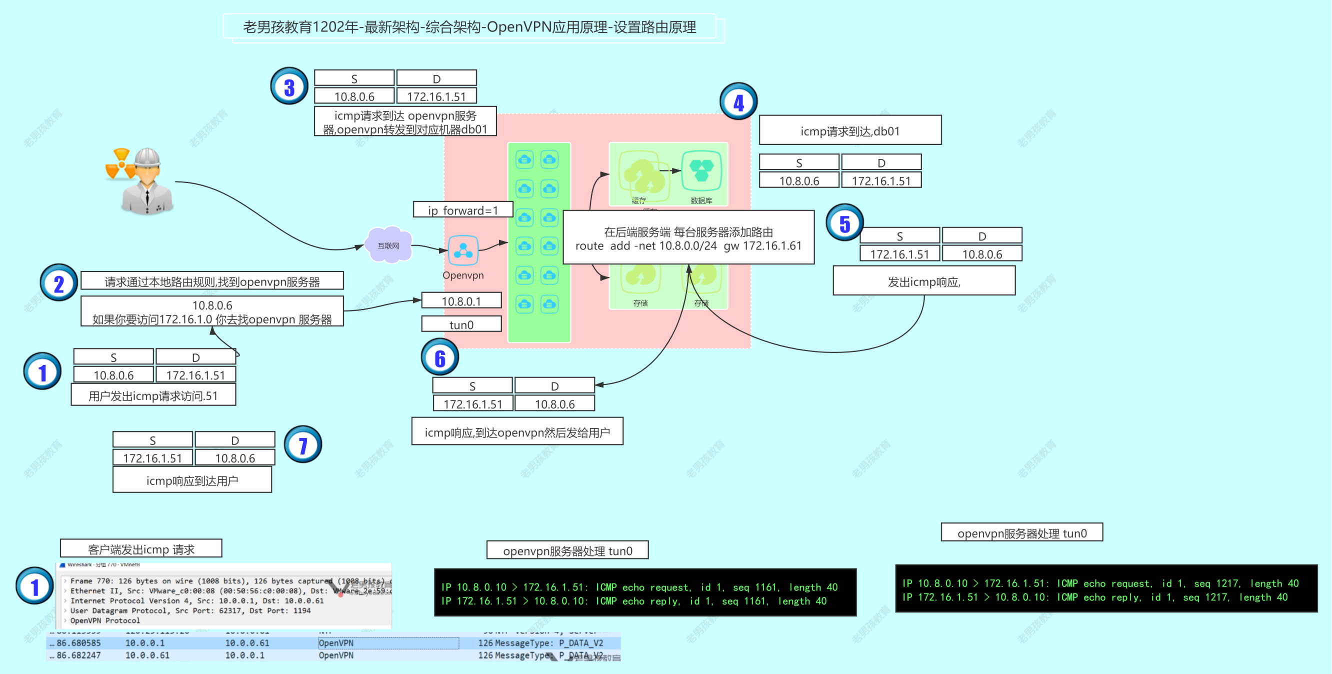


```
[root@m01 /opt/easy-rsa]# tail -f /var/log/openvpn-password.log
2021-06-01 09:54:34: User does not exist: username="lidao", password="1".
2021-06-01 09:55:25: Successful authentication: username="oldboy".
```

7. OpenVPN自动处理内网请求

7.1 目前需要在后端节点 添加路由

高清图解原理图



```
› Frame 770: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits)
› Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_2e:59:
› Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.61
› User Datagram Protocol, Src Port: 62317, Dst Port: 1194
› OpenVPN Protocol
… 86.680585   10.0.0.1    10.0.0.61   OpenVPN   126 MessageType: P_DATA_V2
… 86.682247   10.0.0.61   10.0.0.1    OpenVPN   126 MessageType: P_DATA_V2
```

```
IP 10.8.0.10 > 172.16.1.51: ICMP echo request, id 1, seq 1161, length 40
IP 172.16.1.51 > 10.8.0.10: ICMP echo reply, id 1, seq 1161, length 40
```

```
IP 10.8.0.10 > 172.16.1.51: ICMP echo request, id 1, seq 1217, length 40
IP 172.16.1.51 > 10.8.0.10: ICMP echo reply, id 1, seq 1217, length 40
```

7.2 通过防火墙实现自动回包(需要使用firewalld才行)

- 设置后端节点,网关指向openvpn
- 防火墙

```
[root@db01 ~]# cat  /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=eth0
UUID=de0e3a93-c24e-4031-9a8c-0a212d82d80d
DEVICE=eth0
ONBOOT=no
IPADDR=10.0.0.51
PREFIX=24
GATEWAY=10.0.0.2
DNS1=223.5.5.5
```

```
IPV6_PRIVACY=no
[root@db01 ~]# cat  /etc/sysconfig/network-scripts/ifcfg-eth1
[root@db01 ~]# cat  /etc/sysconfig/network-scripts/ifcfg-eth1
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=eth1
UUID=ce1a7654-0486-49ba-a032-1a071995ce97
DEVICE=eth1
ONBOOT=yes
IPADDR=172.16.1.51
PREFIX=24
GATEWAY=172.16.1.61
DNS1=223.5.5.5
DNS2=223.6.6.6
IPV6_PRIVACY=no

```

- 防火墙规则

```
/usr/share/doc/openvpn-2.4.11/sample/sample-config-files/firewall.sh
iptables -t nat -A POSTROUTING -s 172.16.1.0/24  -j MASQUERADE   #适用于共享上网,公网ip不固定方法
iptables -t nat -A POSTROUTING -s 172.16.1.0/24  -j SNAT --to-source  10.0.0.61 #公网ip

MAS QUE RADE
```

8. 参考与帮助:

中文openvpn传送门

9. 故障记录:

1

```
WARNING: No server certificate verification method has been enabled.  See http://openvpn.net/howto.html#mitm for more info.

 remote-cert-tls server #客户端加上



VERIFY ERROR: depth=1, error=self signed certificate in certificate chain: CN=Easy-RSA CA
```

```
OpenSSL: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed
TLS_ERROR: BIO read tls_read_plaintext error
TLS Error: TLS object -> incoming plaintext read error
TLS Error: TLS handshake failed

看颜色 红色
error
failed

certificate verify failed
证书认证失败.



Tue Jun 01 10:27:41 2021 VERIFY OK: depth=1, CN=Easy-RSA CA
Tue Jun 01 10:27:41 2021 VERIFY OK: depth=0, CN=server
```