

第五次实验报告

课程名称	网络安全实验				
学生姓名	邓鹏	学号	2021302181152	指导老师	陈治宏
专业	网络空间安全	班级	6 班	实验时间	2024-4-24

目录

一、 实验介绍	2
1.1 实验名称	2
1.2 实验任务	2
1.3 实验目的	2
1.4 实验工具	2
1.5 实验环境	2
二、 实验内容	3
2.1 任务一	3
2.1.1 任务描述	3
2.1.2 实验目标	3
2.1.3 实验工具	3
2.1.4 操作步骤	3
2.2 任务二	9
2.2.1 任务描述	9
2.2.2 实验目标	9
2.2.3 实验工具	9
2.2.4 操作步骤	9
2.3 任务三	12
2.3.1 任务描述	12
2.3.2 实验目标	13
2.3.3 实验工具	13
2.3.4 操作步骤	13
2.4 任务四	22
2.4.1 任务描述	22
2.4.2 实验目标	22
2.4.3 实验工具	22
2.4.4 操作步骤	22
2.5 任务五	31
2.5.1 任务描述	31
2.5.2 实验目标	31
2.5.3 实验工具	31
2.5.4 操作步骤	31
三、 实验总结	34

一、实验介绍

1.1 实验名称

VPN 实验

1.2 实验任务

任务一 使用 IP 命令搭建基于隧道的虚拟专有网络

任务二 使用加密工具 OpenSSL 创建加密密钥

任务三 SSL VPN 之 OpenVPN 的安装配置

任务四 IPsecVPN 原理及安装配置

任务五 云计算中基于 Overlay 技术的隧道网络实现

1.3 实验目的

- 掌握如何搭建基于隧道的虚拟专有网络
- 掌握加密算法了解及其应用
- 掌握如何安装部署配置 openvpn 服务端与客户端
- 掌握 IPsecVPN 原理及安装部署
- 了解公有云中 overlay 的实现

1.4 实验工具

- IP
- Openssl
- Ipsec
- ovs-vsctl
- openvpn
- tcpdump
- sysctl
- modprobe
- iptables

1.5 实验环境

操作系统	IP 地址	服务器角色	登录账户密码
Windows2012	192.168.0.11	操作机	用户名：administrator；密码：Simplexue123
centos7_1	192.168.1.11	目标机	用户名：root；密码：Simplexue123
centos7_2	192.168.2.11	目标机	用户名：administrator；密码：Simplexue123

二、实验内容

2.1 任务一

2.1.1 任务描述

实现两不同网络内的内网通过 ip 隧道使之互通并检测。

2.1.2 实验目标

- 了解企业网络环境如何使不同网络之间内网互通。
- 掌握 ip 命令的使用。
- 掌握虚拟私有网络实现方法。

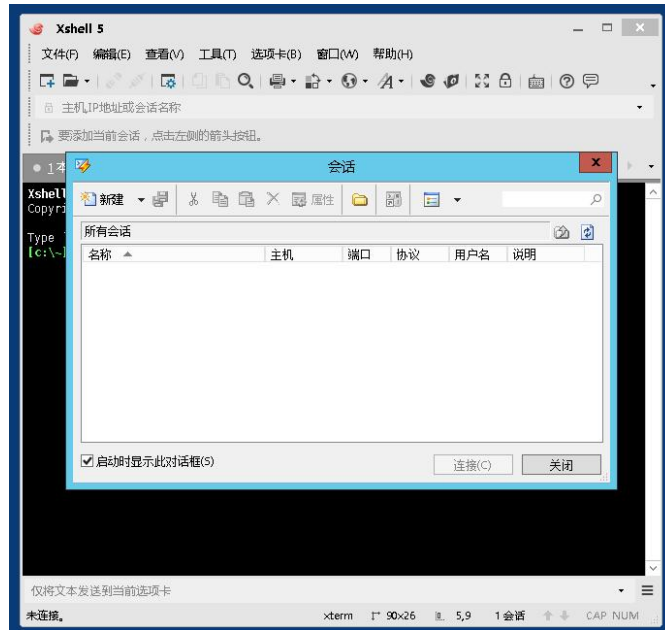
2.1.3 实验工具

- ip
- modprobe

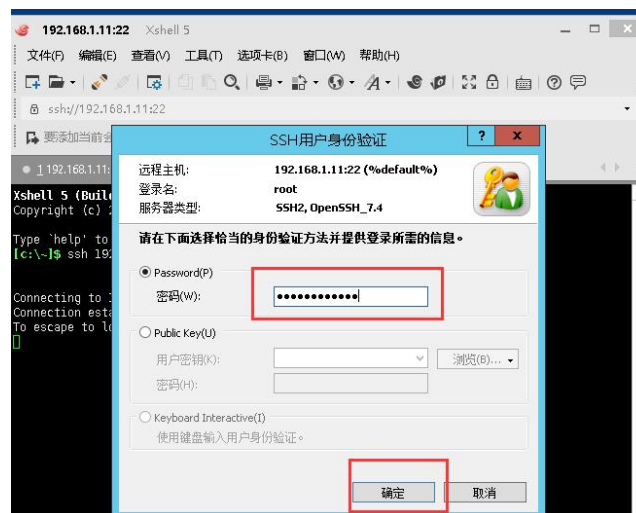
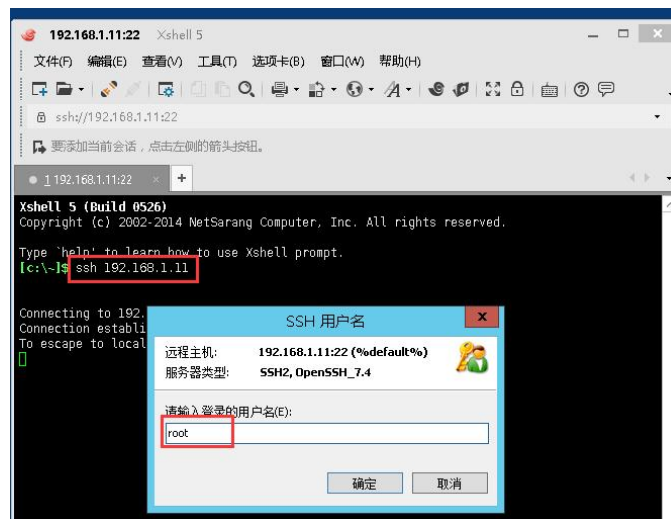
2.1.4 操作步骤

1. 双击桌面 Xshell5 图标，在弹出的界面登陆主机 192.168.1.11 和 192.168.2.11 这两台主机.密码为 Simplexue123

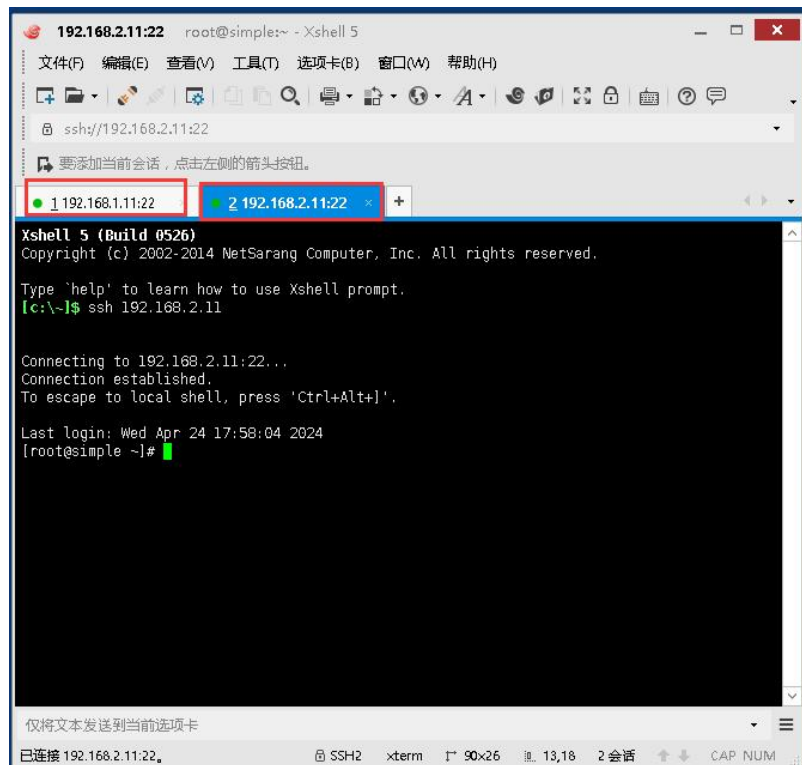
打开 xshell 软件



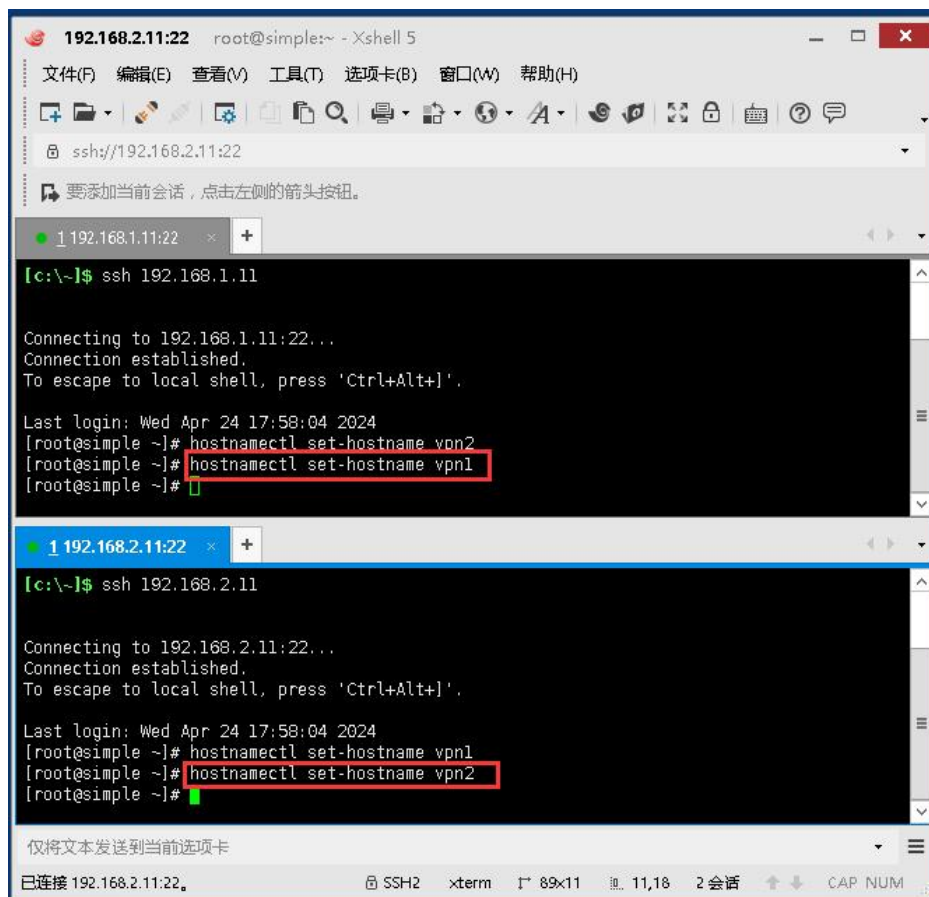
输入命令，在弹窗中输入账号密码



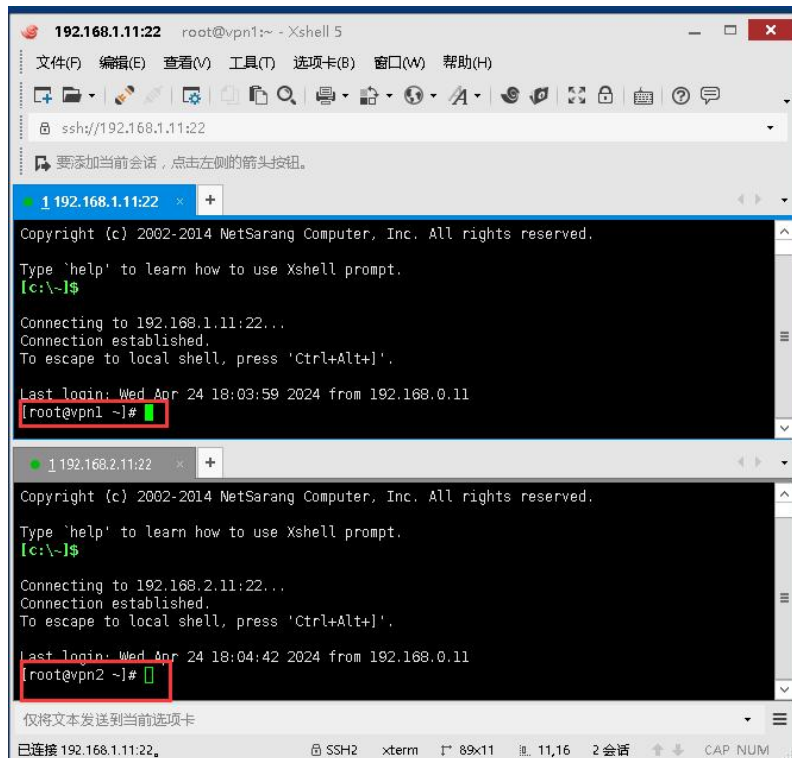
得到如下



- 分别修改主机名：



重新登录，主机名改变：



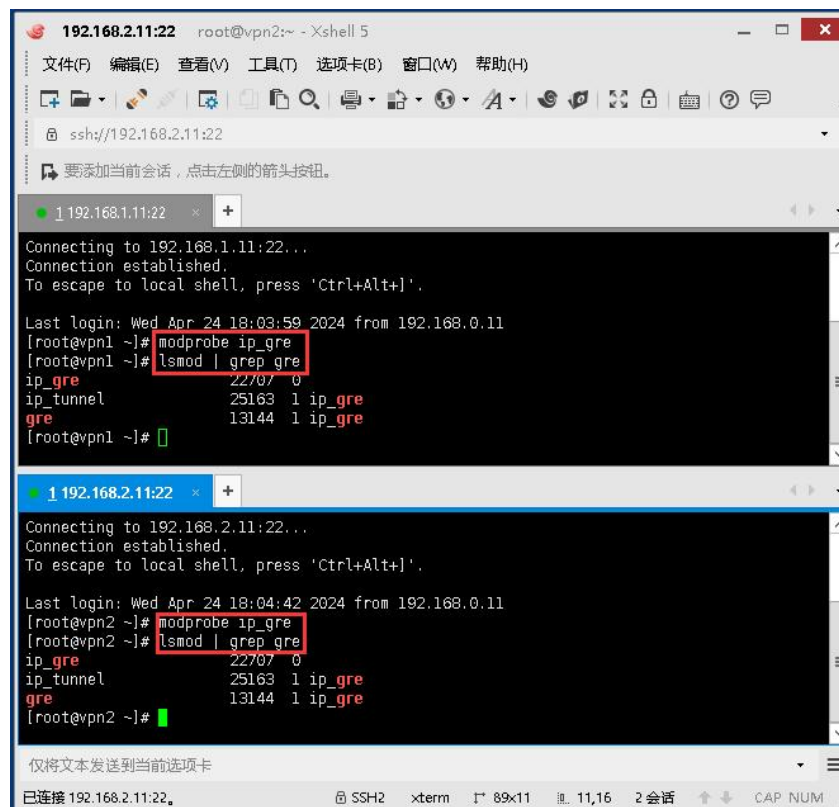
2.vpn1 和 vpn2 主机分别加载 gre 内核模块并检查

加载 ip_gre 内核模块

[root@vpn1 ~]# modprobe ip_gre

查询 ip_gre 模块是否加载，如图所示已正常加载

[root@vpn1 ~]# lsmod | grep gre



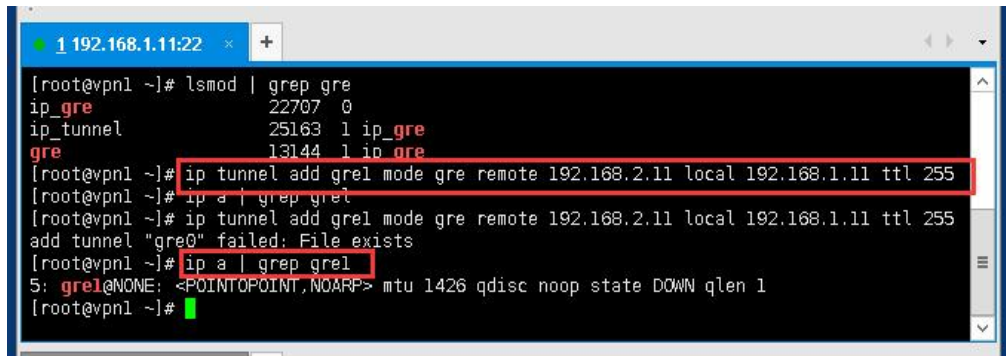
3. 配置 tunnel（GRE 隧道）使它们互通

vpn1 创建一个 GRE 类型隧道设备 gre1，并设置对端 IP 为 192.168.2.11。隧道数据包将被从 192.168.1.11 也就是本地 IP 地址发起，其 TTL 字段被设置为 255。隧道设备分配的 IP 地址为 10.10.10.1，掩码为 255.255.255.0。

- 创建 GRE 类型隧道设备 gre1，并验证是否添加成功

```
[root@vpn1 ~] ip tunnel add gre1 mode gre remote 192.168.2.11 local 192.168.1.11 ttl 255
```

```
[root@vpn1 ~] ip a | grep gre1
```



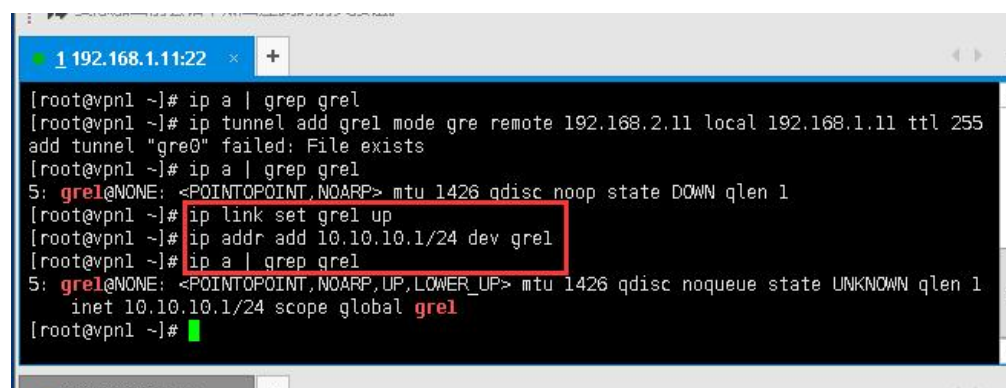
```
[root@vpn1 ~]# lsmod | grep gre
ip_gre                22707  0
ip_tunnel              25163  1 ip_gre
gre                    13144  1 ip_gre
[root@vpn1 ~]# ip tunnel add gre1 mode gre remote 192.168.2.11 local 192.168.1.11 ttl 255
[root@vpn1 ~]# ip a | grep gre1
[root@vpn1 ~]# ip tunnel add gre1 mode gre remote 192.168.2.11 local 192.168.1.11 ttl 255
add tunnel "gre0" failed: File exists
[root@vpn1 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP> mtu 1426 qdisc noop state DOWN qlen 1
[root@vpn1 ~]#
```

- 启动 gre1 并分配 ip 地址 10.10.10.1，检测是否添加并启动。

```
# ip link set gre1 up
```

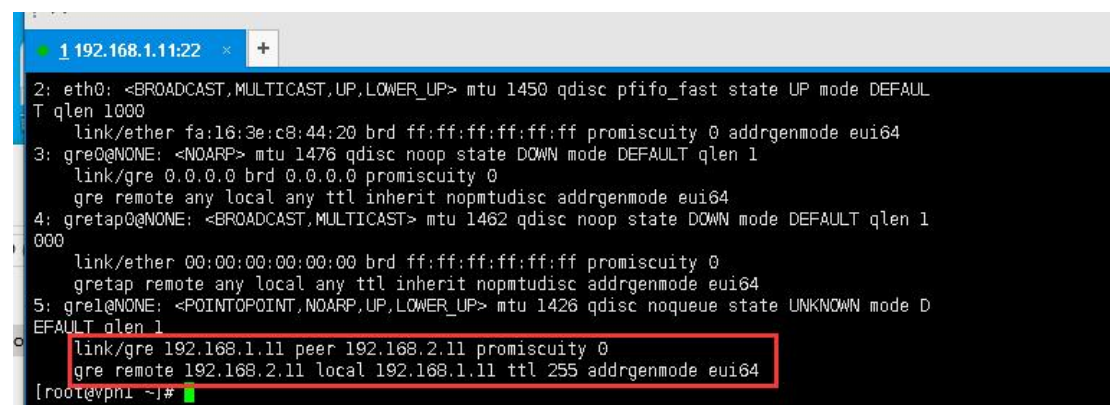
```
# ip addr add 10.10.10.1/24 dev gre1
```

```
# ip a | grep gre1
```



```
[root@vpn1 ~]# ip a | grep gre1
[root@vpn1 ~]# ip tunnel add gre1 mode gre remote 192.168.2.11 local 192.168.1.11 ttl 255
add tunnel "gre0" failed: File exists
[root@vpn1 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP> mtu 1426 qdisc noop state DOWN qlen 1
[root@vpn1 ~]# ip link set gre1 up
[root@vpn1 ~]# ip addr add 10.10.10.1/24 dev gre1
[root@vpn1 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 qdisc noqueue state UNKNOWN qlen 1
    inet 10.10.10.1/24 scope global gre1
[root@vpn1 ~]#
```

- 查看隧道状态

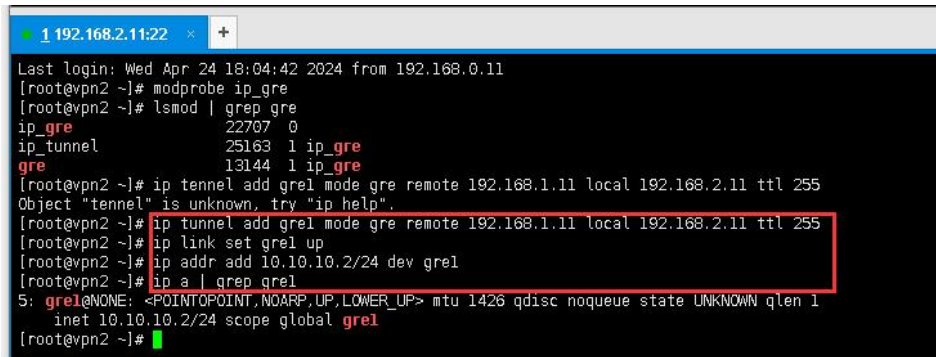


```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc pfifo_fast state UP mode DEFAULT
    qlen 1000
    link/ether fa:16:3e:c8:44:20 brd ff:ff:ff:ff:ff:ff promiscuity 0 addrgenmode eui64
3: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN mode DEFAULT qlen 1
    link/gre 0.0.0.0 brd 0.0.0.0 promiscuity 0
    gre remote any local any ttl inherit nopmtudisc addrgenmode eui64
4: gretap@NONE: <BROADCAST,MULTICAST> mtu 1462 qdisc noop state DOWN mode DEFAULT qlen 1
    qlen 000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff promiscuity 0
    gretap remote any local any ttl inherit nopmtudisc addrgenmode eui64
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 qdisc noqueue state UNKNOWN mode D
    EFAULT qlen 1
    link/gre 192.168.1.11 peer 192.168.2.11 promiscuity 0
    gre remote 192.168.2.11 local 192.168.1.11 ttl 255 addrgenmode eui64
[root@vpn1 ~]#
```


- vpn2 创建一个 GRE 类型隧道设备 gre1, 并设置对端 IP 为 192.168.1.11。隧道数据包将被从 192.168.2.11 也就是本地 IP 地址发起, 其 TTL 字段被设置为 255。隧道设备分配的 IP 地址为 10.10.10.2, 掩码为 255.255.255.0。

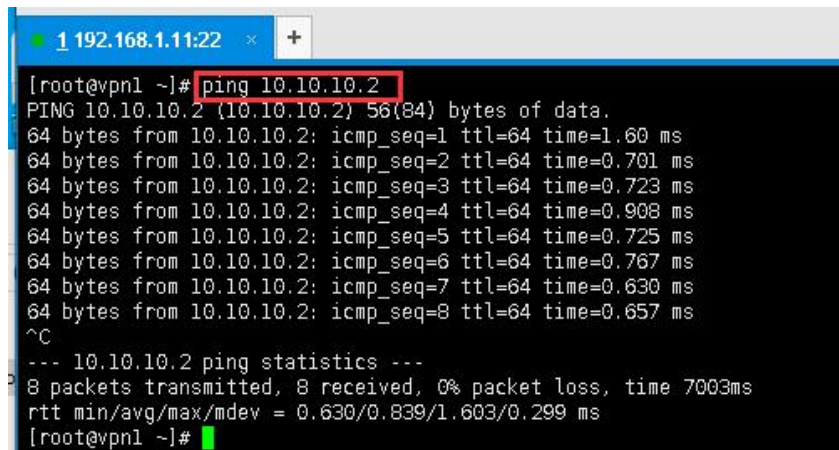
操作步骤如下

```
# ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11 ttl 255
# ip link set gre1 up
# ip addr add 10.10.10.2/24 dev gre1
# ip a | grep gre1
```



```
1 192.168.2.11:22 * +
Last login: Wed Apr 24 18:04:42 2024 from 192.168.0.11
[root@vpn2 ~]# modprobe ip_gre
[root@vpn2 ~]# lsmod | grep gre
ip_gre                22707  0
ip_tunnel             25163  1 ip_gre
gre                   13144  1 ip_gre
[root@vpn2 ~]# ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11 ttl 255
Object "tunnel" is unknown, try "ip help".
[root@vpn2 ~]# ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11 ttl 255
[root@vpn2 ~]# ip link set gre1 up
[root@vpn2 ~]# ip addr add 10.10.10.2/24 dev gre1
[root@vpn2 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 qdisc noqueue state UNKNOWN qlen 1
    inet 10.10.10.2/24 scope global gre1
[root@vpn2 ~]#
```

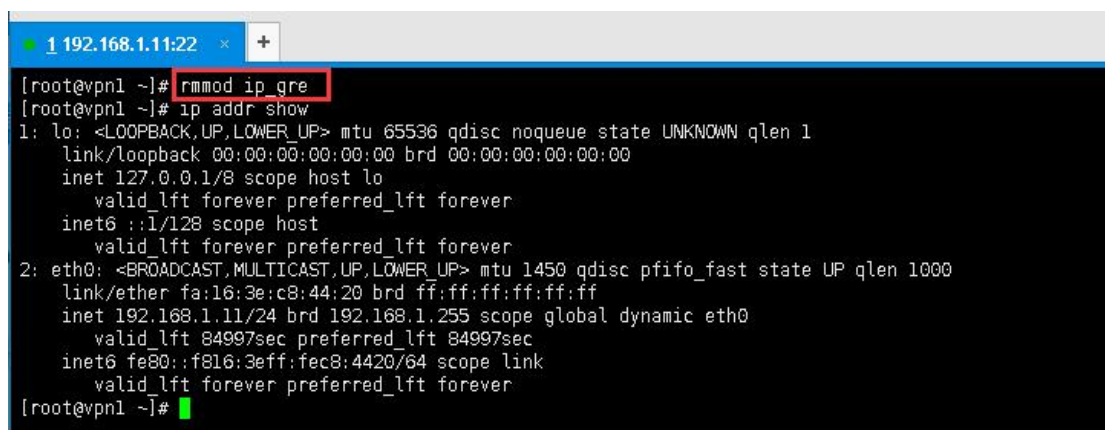
- 测试隧道是否通
ping 检测



```
1 192.168.1.11:22 * +
[root@vpn1 ~]# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data:
 64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=1.60 ms
 64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.701 ms
 64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.723 ms
 64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=0.908 ms
 64 bytes from 10.10.10.2: icmp_seq=5 ttl=64 time=0.725 ms
 64 bytes from 10.10.10.2: icmp_seq=6 ttl=64 time=0.767 ms
 64 bytes from 10.10.10.2: icmp_seq=7 ttl=64 time=0.630 ms
 64 bytes from 10.10.10.2: icmp_seq=8 ttl=64 time=0.657 ms
^C
--- 10.10.10.2 ping statistics ---
 8 packets transmitted, 8 received, 0% packet loss, time 7003ms
 rtt min/avg/max/mdev = 0.630/0.839/1.603/0.299 ms
[root@vpn1 ~]#
```

4. 卸载 GRE 模块

```
# rmmod ip_gre
```



```
1 192.168.1.11:22 * +
[root@vpn1 ~]# rmmod ip_gre
[root@vpn1 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc pfifo_fast state UP qlen 1000
    link/ether fa:16:3e:c8:44:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 84997sec preferred_lft 84997sec
    inet6 fe80::f816:3eff:fec8:4420/64 scope link
        valid_lft forever preferred_lft forever
[root@vpn1 ~]#
```


2.2 任务二

2.2.1 任务描述

本实验主要是用来了解 openssl 的使用及原理，通过本实验可以了解如何实现密钥证书管理、对称加密和非对称加密。

2.2.2 实验目标

- 了解 openssl 加密解密原理。
- 掌握 openssl 如何生成公钥私钥，以及公私钥之间的相互转化。
- 掌握如何用 openssl 生成带密码的公钥私钥，以及之间的加密解密。
- 掌握如何生成带签名信息的证书。

2.2.3 实验工具

- openssl

2.2.4 操作步骤

1. 1.查看 openssl 命令的基本帮助

```
[root@vpn1 ~]# openssl genrsa -
//密钥位数，建议 1024 及以上
usage: genrsa [args] [numbits]
//生成的密钥使用 des 方式进行加密
    -des    encrypt the generated key with DES in cbc mode
//生成的密钥使用 des3 方式进行加密
    -des3   encrypt the generated key with DES in ede cbc mode (168 bit key)
    -idea   encrypt the generated key with IDEA in cbc mode
//生成的密钥还是要 seed 方式进行
    -seed   encrypt PEM output with cbc seed
//生成的密钥使用 aes 方式进行加密
    -aes128, -aes192, -aes256 encrypt PEM output with cbc aes
//生成的密钥使用 camellia 方式进行加密
    -camellia128, -camellia192, -camellia256 encrypt PEM output with cbc camellia
//生成的密钥文件，可从中提取公钥
    -out file    output the key to 'file'
//指定密钥文件的加密口令，可从文件、环境变量、终端等输入
    -passout arg    output file pass phrase source
//选择指数 e 的值，默认指定该项，e 值为 65537
```



```
[root@vpn1 ~]# openssl rsa -in rsa_private.key -pubout -out rsa_public.key
writing RSA key
[root@vpn1 ~]# ll rsa_public.key
-rw-r--r-- 1 root root 451 4月 24 18:28 rsa_public.key
[root@vpn1 ~]#
```

3. 生成 RAS 含密码（使用 aes256 加密）公私钥

其中 passout 代替 shell 进行密码输入，否则会提示输入密码

```
[root@vpn1 ~]# openssl genrsa -aes256 -passout pass:simple -out
rsa_aes_private.key 2048
```

生成其对应的公钥，需要输入密码，其中 pass 代替 shell 进行密码输入，否则会提示输入密码；

```
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -pubout
-out rsa_aes_public.key
```

```
[root@vpn1 ~]# ll rsa_*
```

```
[root@vpn1 ~]# openssl genrsa -aes256 -passout pass:simple -out rsa_aes_private.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
e is 65537 (0x10001)
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -pubout -out rsa_aes_public.key
Invalid password argument "pass"
Error getting passwords
-bash: simple: 未找到命令
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -pubout -out rsa_aes_public.key
Invalid password argument "pass"
Error getting passwords
-bash: simple: 未找到命令
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -pubout -out rsa_aes_public.key
writing RSA key
[root@vpn1 ~]# ll rsa_*
-rw-r--r-- 1 root root 1766 4月 24 18:30 rsa_aes_private.key
-rw-r--r-- 1 root root 451 4月 24 18:31 rsa_aes_public.key
-rw-r--r-- 1 root root 1679 4月 24 18:26 rsa_private.key
-rw-r--r-- 1 root root 451 4月 24 18:28 rsa_public.key
[root@vpn1 ~]#
```

4. 加密与非加密之间的转换

私钥转非加密

```
openssl rsa -in rsa_aes_private.key -passin pass:simple -out rsa_private.key
```

私钥转加密

```
openssl rsa -in rsa_private.key -aes256 -passout pass:simple -out rsa_aes_private.key
```

```
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -out rsa_private.key
writing RSA key
[root@vpn1 ~]# openssl rsa -in rsa_private.key -aes256 -passout pass:simple -out rsa_aes_private.key
writing RSA key
[root@vpn1 ~]#
```

5. 生成自签名证书

生成 RSA 私钥和自签名证书

req 是证书请求的子命令，-newkey rsa:2048 -keyout private_key.pem 表示生成私钥(PKCS8 格式)，-nodes 表示私钥不加密，若不带参数将提示输入密码；-x509 表示输出证书，-days365 为有效期，此后根据提示输入证书拥有者信息；

```
openssl req -newkey rsa:2048 -nodes -keyout rsa_private.key -x509 -days 365
-out cert.crt
```

若执行自动输入，可使用-subj 选项：

```
openssl req -newkey rsa:2048 -nodes -keyout rsa_private.key -x509 -days 365
-out cert.crt -subj "/C=CN/ST=BJ/L=BJ/O=simpleedu/OU=edu/CN=simple/email"
```

Address=simple@simpleedu.com"

使用已有 RSA 私钥生成自签名证书

-new 指生成证书请求，加上-x509 表示直接输出证书，-key 指定私钥文件，其余选项与上述命令相同

openssl req -new -x509 -days 365 -key rsa_private.key -out cert.crt

根据提示输入相应的信息即可

```
[root@vpn1 ~]# openssl req -newkey rsa:2048 -nodes -keyout rsa_private.key -x509 -days 365 -out cert.crt -subj "/C=CN/ST=BJ/L=BJ/O=simpleedu/OU=edu/CN=simple/emailAddress=simple@simpleedu.com"
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'rsa_private.key'
-----
[root@vpn1 ~]# ll cert.crt rsa_private.key
-rw-r--r-- 1 root root 1389 4月 24 18:37 cert.crt
-rw-r--r-- 1 root root 1704 4月 24 18:37 rsa_private.key
[root@vpn1 ~]#
```

6. 生成签名请求及 CA 签名

使用 RSA 私钥生成 CSR 签名请求

openssl genrsa -aes256 -passout pass:simpleedu -out server.key 2048

openssl req -new -key server.key -out server.csr

* 此时生成的 csr 签名请求文件可提交至 CA 进行签发 *

```
[root@vpn1 ~]# openssl genrsa -aes256 -passout pass:simpleedu -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
e is 65537 (0x10001)
```

```
Email Address [simple@simpleedu.com]:
[root@vpn1 ~]# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
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.
-----
Country Name (2 letter code) [XX]:CN
State or Province Name (full name) []:BJ
Locality Name (eg, city) [Default City]:BJ
Organization Name (eg, company) [Default Company Ltd]:simpleedu
Organizational Unit Name (eg, section) []:simple
Common Name (eg, your name or your server's hostname) []:simpleedu
Email Address []:simple@simpleedu.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:simple
An optional company name []:simple
[root@vpn1 ~]#
```

2.3 任务三

2.3.1 任务描述

本实验任务基于真实企业网络环境，在两台服务器搭建的典型企业局域网环境中，主要完成以下内容：

- (1) 搭建 openvpn 服务端与客户端。
- (2) 实现客户端可访问服务端机器

2.3.2 实验目标

- 了解企业级别 openvpn 的使用场景。
- 掌握企业级别 openvpn 搭建和使用。
- 掌握 openvpn 客户端与服务端的搭建配置。

2.3.3 实验工具

- openvpn

2.3.4 操作步骤

1. 在 vpn1 机器安装 openvpn 并验证

- 修改 yum 源

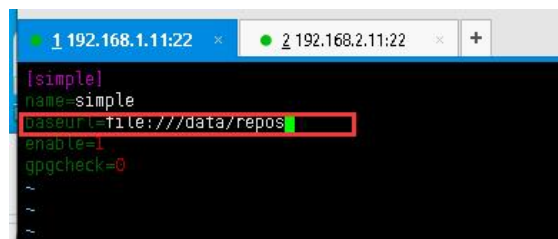
输入: vim /etc/yum.repos.d/simple.repo

将 baseurl=http://192.168.50.50/yum 修改为 baseurl=file:///data/repos

i 进入插入状态, 修改完成后输入:wq 保存退出

(若遇进程问题, 则输入 kill -9 PID, PID 为当前进程)

```
http://192.168.50.50/yum/repodata/repomd.xml?_id=14] Ctrl+#38 - callback abort
[root@vpn1 ~]# vim /etc/yum.repos.d/simple.repo
```



```
[root@vpn1 ~]# yum clean all
```

```
[root@vpn1 ~]# yum install openvpn -y
```

```

[root@vpn1 ~]# yum clean all
已加载插件: fastestmirror
正在清理软件源: simple
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or removed repos
[root@vpn1 ~]# yum install openvpn -y
已加载插件: fastestmirror
simple
simple/primary_db
Determining fastest mirrors
正在解决依赖关系
--> 正在检查事务
--> 软件包 openvpn.x86_64.0.2.4-1.el7 将被 安装
--> 正在处理依赖关系 libpkcs11-helper.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在处理依赖关系 liblz4.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在检查事务
--> 软件包 lz4.x86_64.0.1.7.3-1.el7 将被 安装
--> 软件包 pkcs11-helper.x86_64.0.1.11-3.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决
```



```
[root@vpn1 ~]# rpm -qa | grep openvpn
```

```
完毕!  
[root@vpn1 ~]# rpm -qa | grep openvpn  
openvpn-2.4.4-1.el7.x86_64  
[root@vpn1 ~]#
```

2. 修改 openvpn 的配置文件 server.conf 配置文件的内容如下

- 拷贝模板文件到配置文件目录下

```
[root@vpn1 ~]# cp /usr/share/doc/openvpn-2.4.4/sample/sample-config-files/server.conf /etc/openvpn/
```

```
[root@vpn1 ~]# ls /etc/openvpn/server.conf
```

```
openvpn-2.4.4-1.el7.x86_64  
[root@vpn1 ~]# cp /usr/share/doc/openvpn-2.4.4/sample/sample-config-files/server.conf /etc/openvpn/  
[root@vpn1 ~]# ls /etc/openvpn/server.conf  
/etc/openvpn/server.conf  
[root@vpn1 ~]#
```

- 修改 openvpn 服务端的配置文件/etc/openvpn/server.conf

```
[root@vpn1 ~]# vim /etc/openvpn/server.conf
```

```
/etc/openvpn/server.conf  
[root@vpn1 ~]# vim /etc/openvpn/server.conf
```

指定 TCP 协议(使用 TCP 协议如果连接上 VPN 后网络很慢,可以更改成使用 UDP 协议)

```
# TCP or UDP server?  
proto tcp  
# proto udp
```

打开这三行注释,配置 DNS (实验环境无法连通外网,可不配置)

```
# or bridge the TUN/TAP interface to the internet  
# in order for this to work properly).  
push "redirect-gateway def1 bypass-dhcp"  
  
# Certain Windows-specific network settings  
# can be pushed to clients, such as DNS  
# or WINS server addresses. CAVEAT:  
# http://openvpn.net/faq.html#dhcpcaveats  
# The addresses below refer to the public  
# DNS servers provided by opendns.com  
push "dhcp-option DNS 208.67.222.222"  
push "dhcp-option DNS 208.67.220.220"  
  
# Uncomment this directive to allow different  
# clients to be able to "see" each other.  
# By default, clients will only see the server
```

设置启动用户

```
# You can uncomment this out on  
# non-Windows systems.  
user nobody  
group nobody
```

注释掉 explicit-exit-notify 1

```
# Notify the client that when the server restarts so it
# can automatically reconnect.
#explicit-exit-notify 1
-- 插入 --
```

3. 安装密钥生成软件

```
[root@vpn1 ~]# yum install easy-rsa -y
```

```
[root@vpn1 ~]# vim /etc/openvpn/server.conf
[root@vpn1 ~]# yum install easy-rsa -y
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
--> 软件包 easy-rsa.noarch.0.2.2.2-1.el5 将被 安装
--> 解决依赖关系完成

依赖关系解决

=====
Package                                架构
=====
正在安装:
```

4. 准备配置证书文件

- 拷贝文件到/etc/openvpn

```
[root@vpn1 ~]# cp -r /usr/share/easy-rsa/ /etc/openvpn/
```

```
[root@vpn1 ~]# ls /etc/openvpn/easy-rsa/
```

```
[root@vpn1 ~]# cp -r /usr/share/easy-rsa/ /etc/openvpn/
[root@vpn1 ~]# ls /etc/openvpn/easy-rsa/
2.0
[root@vpn1 ~]#
```

- 配置生成证书的环境变量,并使之生效

```
[root@vpn1 ~]# vim /etc/openvpn/easy-rsa/2.0/vars
```

```
[root@vpn1 ~]# vim /etc/openvpn/easy-rsa/2.0/vars
[root@vpn1 ~]#
```

\# 现只修改如下几条,可根据自己情况进行修改

```
export KEY_COUNTRY="CN"
export KEY_PROVINCE="BJ"
export KEY_CITY="BEIJING"
export KEY_ORG="SimpleEdu"
export KEY_EMAIL="simpleedu@simple.com"
export KEY_OU="MyOrganizationalUnit"
```

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="CN"
export KEY_PROVINCE="BJ"
export KEY_CITY="BEIJING"
export KEY_ORG="SimpleEdu"
export KEY_EMAIL="simpleedu@simple.com"
export KEY_OU="MyOrganizationalUnit"
```


使配置的环境变量生效

```
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0/
```

```
[root@vpn1 2.0]# source vars
```

```
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0/
[root@vpn1 2.0]# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 2.0]#
```

- 根据提示先删除所有，再根据自己情况进行修改（默认回车即可）

```
[root@vpn1 2.0]# cd /etc/openvpn/easy-rsa/2.0/
```

```
[root@vpn1 2.0]# source vars NOTE: If you run ./clean-all, I will be doing a rm
-rf on /etc/openvpn/easy-rsa/2.0/keys
```

```
[root@vpn1 2.0]# ./clean-all
```

```
[root@vpn1 2.0]# ./build-ca ``
```

```
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0/vars
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0/
[root@vpn1 2.0]# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 2.0]# ./clean-all
[root@vpn1 2.0]# ./build-ca
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca.key'
-----
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.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [BJ]:
Locality Name (eg, city) [BEIJING]:
Organization Name (eg, company) [SimpleEdu]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [SimpleEdu CA]:
Name [EasyRSA]:
Email Address [simpleedu@simple.com]:
[root@vpn1 2.0]#
[root@vpn1 2.0]#
```

5. 建服务端的证书 创建通用名(common name)为”server”的证书文件,交互输入自己的值,回车键进行，在提示输入密码的地方，设置一个密码如 simple123

```
[root@vpn1 2.0]# ./build-key-server server
```

```
[root@vpn1 2.0]# ./build-key-server server
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'server.key'
-----
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.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [BJ]:
Locality Name (eg, city) [BEIJING]:
Organization Name (eg, company) [SimpleEdu]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [server]:
Name [EasyRSA]:
Email Address [simpleedu@simple.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```


- 创建一个通用名(common name)为 client 的客户端证书，交互输入自己的值，默认回车键进行

```
[root@vpn1 keys]# cd ..
```

```
[root@vpn1 2.0]# ./build-key client
```

```
[root@vpn1 2.0]# ll keys/client.*
```

```
[root@vpn1 keys]# cd ..
[root@vpn1 2.0]# ./build-key client
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'client.key'
-----
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.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [BJ]:
Locality Name (eg, city) [BEIJING]:
Organization Name (eg, company) [SimpleEdu]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [client]:
Name [EasyRSA]:
Email Address [simpleedu@simple.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openssl/easy-rsa/2.0/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName               :PRINTABLE:'CN'
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openssl/easy-rsa/2.0/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName               :PRINTABLE:'CN'
stateOrProvinceName       :PRINTABLE:'BJ'
localityName               :PRINTABLE:'BEIJING'
organizationName           :PRINTABLE:'SimpleEdu'
organizationalUnitName     :PRINTABLE:'MyOrganizationalUnit'
commonName                 :PRINTABLE:'client'
name                       :PRINTABLE:'EasyRSA'
emailAddress               :IASSTRING:'simpleedu@simple.com'
Certificate is to be certified until Apr 22 11:14:32 2034 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
```

```
Data Base Updated
[root@vpn1 2.0]# ll keys/client.*
-rw-r--r-- 1 root root 5452 4月 24 19:14 keys/client.crt
-rw-r--r-- 1 root root 1090 4月 24 19:14 keys/client.csr
-rw----- 1 root root 1704 4月 24 19:14 keys/client.key
[root@vpn1 2.0]#
```

7. 启动并检查

- 启动 openvpn 服务并设置为开机自启动

启动 openvpn 服务

```
[root@vpn1 ~]# systemctl start openvpn@server.service
```

设置开机自启动

```
[root@vpn1 ~]# systemctl enable openvpn@server.service
```

查看状态

```
[root@vpn1 ~]# systemctl status openvpn@server.service
```

检查是否启动

```
[root@vpn1 ~]# netstat -lntup | grep openvpn
```

如下所示表示正常启动

```
tcp 0 0 0.0.0.0:1194 0.0.0.0:* LISTEN 8870/openvpn
```

```
[root@vpn1 ~]# cd -
[root@vpn1 ~]# systemctl start openvpn@server.service
[root@vpn1 ~]# systemctl enable openvpn@server.service
Created symlink from /etc/systemd/system/multi-user.target.wants/openvpn@server.service to /usr/lib/systemd/system/openvpn@server.service.
[root@vpn1 ~]# systemctl status openvpn@server.service
● openvpn@server.service - OpenVPN Robust And Highly Flexible Tunneling Application On server
   Loaded: loaded (/usr/lib/systemd/system/openvpn@server.service; enabled; vendor preset: disabled)
   Active: active (running) since 2024-04-24 19:16:43 CST; 26s ago
     Main PID: 12326 (openvpn)
    Status: "Initialization Sequence Completed"
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─12326 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf

4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 Listening for incoming TCP connection on [AF_INET][undef]:1194
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 TCPv4 SERVER link local (bound): [AF_INET][undef]:1194
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 TCPv4 SERVER link remote: [AF_UNSPEC]
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 GID set to nobody
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 UID set to nobody
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 MULTI: multi_init called, r=256 v=256
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 IFCONFIG POOL LIST
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 MULTI: TCP INIT maxclients=1024 maxevents=1028
4月 24 19:16:43 vpn1 openvpn[12326]: Wed Apr 24 19:16:43 2024 Initialization Sequence Completed
[root@vpn1 ~]# netstat -lntup | grep openvpn
tcp 0 0 0.0.0.0:1194 0.0.0.0:* LISTEN 12326/openvpn
[root@vpn1 ~]#
```

8. 客户端 (vpn2) 登录测试

- 在客户端安装 openvpn

```
[root@vpn2 ~]# yum install openvpn -y
```

```
[root@vpn2 ~]# vim /etc/yum.repos.d/simple.repo
[root@vpn2 ~]#
```

```
[simple]
name=Simple
baseurl=file:///data/repos
enable=1
gpgcheck=0
```

```
[root@vpn2 ~]# vim /etc/yum.repos.d/simple.repo
[root@vpn2 ~]# yum install openvpn -y
已加载插件: fastestmirror
simple
simple/primary_db
Determining fastest mirrors
正在解决依赖关系
--> 正在检查事务
--> 软件包 openvpn.x86_64.0.2.4-1.el7 将被安装
--> 正在处理依赖关系 libpkcs11-helper.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在处理依赖关系 liblz4.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在检查事务
--> 软件包 lz4.x86_64.0.1.7.3-1.el7 将被安装
--> 软件包 pkcs11-helper.x86_64.0.1.11-3.el7 将被安装
--> 解决依赖关系完成
```


- 在 vpn1 端把生产文件拷贝到客户端

```
[root@vpn1 keys]# cd /etc/openvpn/easy-rsa/2.0/keys/
```

\\# 密码为 Simplexue123

```
[root@vpn1 keys]# scp ca.crt client.crt client.key ta.key 192.168.2.11:/etc/openvpn/client/``
```

```
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 keys]# scp ca.crt client.crt client.key ta.key 192.168.2.11:/etc/openvpn/client/
The authenticity of host '192.168.2.11 (192.168.2.11)' can't be established.
ECDSA key fingerprint is SHA256:UEK20E1c33prBueqya21Pv04jd1nsS21PhgXEEI.
ECDSA key fingerprint is MD5:el:8a:eb:50:f4:04:66:79:ab:59:fa:28:02:f5:ca:40.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.11' (ECDSA) to the list of known hosts.
root@192.168.2.11's password:
Permission denied, please try again.
root@192.168.2.11's password:
ca.crt 100% 1781 3.5MB/s 00:00
client.crt 100% 5432 7.2MB/s 00:00
client.key 100% 1704 2.5MB/s 00:00
ta.key 100% 636 1.7MB/s 00:00
[root@vpn1 keys]#
```

- 编辑客户端配置文件

```
[root@vpn2 ~]# vim /etc/openvpn/client/client.conf
```

```
[root@vpn2 client]# vim /etc/openvpn/client/client.conf
[root@vpn2 client]# cat /etc/openvpn/client/client.conf

client
dev tun
proto tcp
remote 192.168.1.11 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/client.crt
key /etc/openvpn/client/client.key
tls-auth /etc/openvpn/client/ta.key 1
cipher AES-256-CBC
verb 3
mute 20
```

- 启动 openvpn 客户端并挂后台运行，并可实时查看其日志。

```
[root@vpn2 client]# cd /etc/openvpn/client/
```

```
[root@vpn2 client]# openvpn /etc/openvpn/client/client.conf &
```

```
mute 20
[root@vpn2 client]# cd /etc/openvpn/client/
[root@vpn2 client]# openvpn /etc/openvpn/client/client.conf &
[1] 12251
```

```
Wed Apr 24 19:42:25 2024 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass-dhcp,dhcp-option DNS 208.67.222.222,dhcp-option DNS 2
t30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM'
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: timers and/or timeouts modified
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: --ifconfig/up options modified
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: route options modified
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: peer-id set
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: adjusting link_mtu to 1626
Wed Apr 24 19:42:25 2024 OPTIONS IMPORT: data channel crypto options modified
Wed Apr 24 19:42:25 2024 Data Channel: using negotiated cipher 'AES-256-GCM'
Wed Apr 24 19:42:25 2024 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Wed Apr 24 19:42:25 2024 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Wed Apr 24 19:42:25 2024 ROUTE GATEWAY 192.168.2.1/255.255.255.0 IFACE=eth0 HWADDR=fa:16:3e:e8:0e:ba
Wed Apr 24 19:42:25 2024 TUN/TAP device tun0 opened
Wed Apr 24 19:42:25 2024 TUN/TAP TX queue length set to 100
Wed Apr 24 19:42:25 2024 do_ifconfig, tt->did_ifconfig ipv6 setup=0
Wed Apr 24 19:42:25 2024 /sbin/ip link set dev tun0 up mtu 1500
Wed Apr 24 19:42:25 2024 /sbin/ip addr add dev tun0 local 10.8.0.6 peer 10.8.0.5
Wed Apr 24 19:42:25 2024 /sbin/ip route add 192.168.1.11/32 via 192.168.2.1
Wed Apr 24 19:42:25 2024 /sbin/ip route add 0.0.0.0/1 via 10.8.0.5
Wed Apr 24 19:42:25 2024 /sbin/ip route add 128.0.0.0/1 via 10.8.0.5
Wed Apr 24 19:42:25 2024 /sbin/ip route add 10.8.0.1/32 via 10.8.0.5
Wed Apr 24 19:42:25 2024 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Wed Apr 24 19:42:25 2024 Initialization Sequence Completed
```

- 查看网卡信息，得知已获取到 ip

```
[root@vpn2 ~]# ip addr show tun0
```

```
[root@vpn2 client]# ip addr show tun0
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 100
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::e23b:b517:dc29:3592/64 scope link flags 800
        valid_lft forever preferred_lft forever
[root@vpn2 client]#
```

- 测试是否可使用

```
[root@vpn2 client]# ping 10.8.0.1
```

```
[root@vpn2 client]# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data:
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.854 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.957 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=1.36 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=1.09 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=0.896 ms
^C
--- 10.8.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.854/1.032/1.361/0.187 ms
[root@vpn2 client]#
```

- openvpn nat 配置

```
[root@vpn1 ~]# iptables -t nat -A POSTROUTING -s 10.8.0.1/24 -j MASQUERADE
```

ADE

```
[root@vpn1 keys]# cd ~
[root@vpn1 ~]# iptables -t nat -A POSTROUTING -s 10.8.0.1/24 -j MASQUERADE
[root@vpn1 ~]# iptables -t nat -nvL
Chain PREROUTING (policy ACCEPT 1 packets, 94 bytes)
  pkts bytes target     prot opt in     out     source    destination
Chain INPUT (policy ACCEPT 1 packets, 94 bytes)
  pkts bytes target     prot opt in     out     source    destination
Chain OUTPUT (policy ACCEPT 1 packets, 124 bytes)
  pkts bytes target     prot opt in     out     source    destination
Chain POSTROUTING (policy ACCEPT 1 packets, 124 bytes)
  pkts bytes target     prot opt in     out     source    destination
  0      0 MASQUERADE all  --  *      *        10.8.0.0/24  0.0.0.0/0
[root@vpn1 ~]#
```

在 vpn2 上测试

```
[root@vpn2 ~]# ping -c 1 www.baidu.com ``
```

```
[root@vpn2 client]# ping -c 1 www.baidu.com
ping: www.baidu.com: 域名解析暂时失败
[root@vpn2 client]#
```

实验环境不能外网，所以失败

- 关闭服务

```
[root@vpn1 ~]# pkill openvpn
```

```
[root@vpn2 ~]# pkill openvpn
```

```
[root@vpn1 ~]# pkill openvpn
[root@vpn1 ~]#
```

```
[root@vpn2 client]# pkill openvpn  
[root@vpn2 client]#
```

2.4 任务四

2.4.1 任务描述

本实验任务基于真实企业网络环境，在两台服务器搭建的典型企业局域网环境中，主要完成以下内容：

- (1) 搭建 ipsec 服务端与客户端。
- (2) 实现客户端可访问服务端机器

2.4.2 实验目标

- 了解企业级别 ipsec 的使用场景。
- 掌握企业级别 ipsec 搭建和使用。
- 掌握 ipsec 客户端与服务端的搭建配置。
- 掌握 ipsec 多种验证方式的实现。

2.4.3 实验工具

- ipsec
- openssl

2.4.4 操作步骤

1. 调整内核参数，开启数据转发，关闭 icmp 重定向并使之生效。（VPN1 和 VPN2 机器都要做）

```
# 将下面配置文件加入/etc/sysctl.conf  
[root@vpn1 ~]# vim /etc/sysctl.conf  
net.ipv4.ip_forward = 1  
net.ipv4.conf.default.rp_filter = 0  
net.ipv4.conf.all.accept_redirects = 0  
net.ipv4.conf.all.send_redirects = 0  
net.ipv4.conf.default.accept_redirects = 0  
net.ipv4.conf.default.send_redirects = 0  
net.ipv4.conf.eth0.accept_redirects = 0  
net.ipv4.conf.eth0.send_redirects = 0  
net.ipv4.conf.eth1.accept_redirects = 0  
net.ipv4.conf.eth1.send_redirects = 0  
net.ipv4.conf.lo.accept_redirects = 0
```



```
net.ipv4.conf.lo.send_redirects = 0
```

```
# 使配置生效
```

```
[root@vpn1 ~]# sysctl -p
```

```
[root@vpn1 ~]# vim /etc/sysctl.conf
[root@vpn1 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/accept_redirects: 没有那个文件或目录
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/send_redirects: 没有那个文件或目录
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
[root@vpn1 ~]#
```

```
[root@vpn2 ~]# vim /etc/sysctl.conf
[root@vpn2 ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/accept_redirects: 没有那个文件或目录
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/send_redirects: 没有那个文件或目录
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
[root@vpn2 ~]#
```

2. 安装 openswan、libreswan 并验证安装。（VPN1 和 VPN2 机器都要做）

- 安装并验证，两台机器都做

```
[root@vpn1 ~]# yum install openswan libreswan -y
```

```
[root@vpn1 ~]# ipsec --version
```

```
Linux Libreswan U3.20/K(no kernel code presently loaded) on 3.10.0-693.5.2.el7.x86_64
```

```
[root@vpn1 ~]# yum install openswan libreswan -y
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
--> 软件包 libreswan.x86_64.0.3.20-5.el7_4 将被 安装
--> 正在处理依赖关系 libunbound.so.2()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在处理依赖关系 libevent_threads-2.0.so.5()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在处理依赖关系 libevent-2.0.so.5()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在检查事务
--> 软件包 libevent.x86_64.0.2.0.21-4.el7 将被 安装
--> 软件包 unbound-libs.x86_64.0.1.4.20-34.el7 将被 安装
```

```
完毕！
[root@vpn1 ~]# ipsec --version
Linux Libreswan U3.20/K(no kernel code presently loaded) on 3.10.0-693.5.2.el7.x86_64
[root@vpn1 ~]#
```

```
net.ipv4.conf.lo.send_redirects = 0
[root@vpn2 ~]# yum install openswan libreswan -y
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
--> 软件包 libreswan.x86_64.0.3.20-5.el7_4 将被 安装
--> 正在处理依赖关系 libunbound.so.2()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在处理依赖关系 libevent_threads-2.0.so.5()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在处理依赖关系 libevent-2.0.so.5()(64bit)，它被软件包 libreswan-3.20-5.el7_4.x86_64 需要
--> 正在检查事务
--> 软件包 libevent.x86_64.0.2.0.21-4.el7 将被 安装
--> 软件包 unbound-libs.x86_64.0.1.4.20-34.el7 将被 安装
```

```

[root@vpn2 ~]# ipsec --version
Linux Libreswan U3.20/K(no kernel code presently loaded) on 3.10.0-693.5.2.el7.x86_64

```

• 启动服务看是否正常，显示如图测正常，若不是请检查内核配置文件，两台机器都验证。

```
[root@vpn1 ~]# systemctl start ipsec.service
```

```
[root@vpn1 ~]# ipsec verify
```

```

[root@vpn1 ~]# systemctl start ipsec.service
[root@vpn1 ~]# ipsec verify
Verifying installed system and configuration files

Version check and ipsec on-path [OK]
Libreswan 3.20 (netkey) on 3.10.0-693.5.2.el7.x86_64
Checking for IPsec support in kernel [OK]
  NETKEY: Testing XFRM related proc values
    ICMP default/send_redirects [OK]
    ICMP default/accept_redirects [OK]
    XFRM larval drop [OK]
Pluto ipsec.conf syntax [OK]
Two or more interfaces found, checking IP forwarding [OK]
Checking rp_filter [ENABLED]
  /proc/sys/net/ipv4/conf/all/rp_filter [ENABLED]
  rp_filter is not fully aware of IPsec and should be disabled
Checking that pluto is running [OK]
  Pluto listening for IKE on udp 500 [OK]
  Pluto listening for IKE/NAT-T on udp 4500 [OK]
  Pluto ipsec.secret syntax [OK]
Checking 'ip' command [OK]
Checking 'iptables' command [OK]
Checking 'prelink' command does not interfere with FIPS [OK]
Checking for obsolete ipsec.conf options [OK]

ipsec verify: encountered 3 errors - see 'man ipsec_verify' for help
[root@vpn1 ~]#

```

```

Linux Libreswan U3.20/K(no kernel code presently loaded) on 3.10.0-693.5.2.el7.x86_64
[root@vpn2 ~]# systemctl start ipsec.service
[root@vpn2 ~]# ipsec verify
Verifying installed system and configuration files

Version check and ipsec on-path [OK]
Libreswan 3.20 (netkey) on 3.10.0-693.5.2.el7.x86_64
Checking for IPsec support in kernel [OK]
  NETKEY: Testing XFRM related proc values
    ICMP default/send_redirects [OK]
    ICMP default/accept_redirects [OK]
    XFRM larval drop [OK]
Pluto ipsec.conf syntax [OK]
Two or more interfaces found, checking IP forwarding [OK]
Checking rp_filter [ENABLED]
  /proc/sys/net/ipv4/conf/all/rp_filter [ENABLED]
  /proc/sys/net/ipv4/conf/gre0/rp_filter [ENABLED]
  /proc/sys/net/ipv4/conf/gre1/rp_filter [ENABLED]
  /proc/sys/net/ipv4/conf/gretap0/rp_filter [ENABLED]
  rp_filter is not fully aware of IPsec and should be disabled
Checking that pluto is running [OK]
  Pluto listening for IKE on udp 500 [OK]
  Pluto listening for IKE/NAT-T on udp 4500 [OK]
  Pluto ipsec.secret syntax [OK]
Checking 'ip' command [OK]
Checking 'iptables' command [OK]
Checking 'prelink' command does not interfere with FIPS [OK]
Checking for obsolete ipsec.conf options [OK]

ipsec verify: encountered 9 errors - see 'man ipsec_verify' for help
[root@vpn2 ~]#

```

```
[root@vpn1 ~]# netstat -lntup | grep pluto
```

openswan 监听在 UDP 的 500 和 4500 两个端口，其中 500 是用来 IKE 密钥

交换协商，4500 的 NAT-T 是 nat 穿透的

```
ipsec verify: encountered 9 errors - see 'man ipsec_verify' for help
[root@vpn1 ~]# netstat -lntup | grep pluto
udp        0      0 0.0.0.0:12978 0.0.0.0:*           12978/pluto
udp        0      0 0.0.0.0:12978 0.0.0.0:*           12978/pluto
udp        0      0 0.0.0.0:12978 0.0.0.0:*           12978/pluto
udp        0      0 0.0.0.0:12978 0.0.0.0:*           12978/pluto
udp6       0      0 :::12978 :::*           12978/pluto
[root@vpn1 ~]#

ipsec verify: encountered 9 errors - see 'man ipsec_verify' for help
[root@vpn2 ~]# netstat -lntup | grep pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp        0      0 0.0.0.0:12859 0.0.0.0:*           12859/pluto
udp6       0      0 :::12859 :::*           12859/pluto
[root@vpn2 ~]#
```

3. 配置 ipsecVPN 配置（模式为 network-to-network），下面介绍两种认证方式

- 基于 pre-shared keys 认证方式（PSK）

配置/etc/ipsec.conf 配置文件末尾增加如下（VPN1 和 VPN2 的配置附件相同）

```
conn net-to-net
ike=aes256-sha2_256;modp2048
phase2alg=aes256-sha2_256;modp2048
# 使用预共享密钥方式进行认证
authby=secret
type=tunnel
# 一端 IP 地址
left=192.168.1.11
# 一端内网网段地址
leftsubnet=10.0.0.0/24
# 一端的标识符，可以任意填写，如果多个连接需要区分
leftid=@vpn1
leftnexthop=%defaulttroute
right=192.168.2.11
rightsubnet=10.0.1.0/24
rightid=@vpn2
rightnexthop=%defaulttroute
# add 代表只是添加，但并不会连接，如果为 start 则代表着启动自动连接
auto=add
```

```
[root@vpn1 ~]# vim /etc/ipsec.conf
[root@vpn1 ~]# vim /etc/ipsec.conf
[root@vpn1 ~]#
```

```
conn net-to-net
    ike=aes256-sha2_256;modp2048
    phase2alg=aes256-sha2_256;modp2048
    authby=secret
    type=tunnel
    left=192.168.1.11
    leftsubnet=10.0.0.0/24
    leftid=@vpn1
    leftnexthop=%defaultroute
    right=192.168.2.11
    rightsubnet=10.0.1.0/24
    rightid=@vpn2
    rightnexthop=%defaultroute
    auto=add
"/etc/ipsec.conf" 67L, 2343C
```

```
[root@vpn2 ~]# vim /etc/ipsec.conf
[root@vpn2 ~]#
```

```
conn net-to-net
    ike=aes256-sha2_256;modp2048
    phase2alg=aes256-sha2_256;modp2048
    authby=secret
    type=tunnel
    left=192.168.1.11
    leftsubnet=10.0.0.0/24
    leftid=@vpn1
    leftnexthop=%defaultroute
    right=192.168.2.11
    rightsubnet=10.0.1.0/24
    rightid=@vpn2
    rightnexthop=%defaultroute
    auto=add
-- 输入 --
```

- 两台机器是基于密码来配置的，修改 VPN1 和 VPN2 的密码配置文件，分别如下

VPN1 如下

```
[root@vpn1 ~]# cat /etc/ipsec.secrets
include /etc/ipsec.d/*.secrets
192.168.1.11 %any 0.0.0.0 : PSK "123"
```

VPN2 如下

```
[root@vpn2 ~]# cat /etc/ipsec.secrets
include /etc/ipsec.d/*.secrets
192.168.2.11 %any 0.0.0.0 : PSK "123"
```

```
[root@vpn1 ~]# vim /etc/ipsec.secrets
[root@vpn1 ~]# cat /etc/ipsec.secrets
include /etc/ipsec.d/*.secrets
192.168.1.11 %any 0.0.0.0 : PSK "123"
```

```
[root@vpn2 ~]# vim /etc/ipsec.secrets
[root@vpn2 ~]# cat /etc/ipsec.secrets
include /etc/ipsec.d/*.secrets
192.168.2.11 %any 0.0.0.0 : PSK "123"
```


- 两端重新启动服务，并验证。

VPN1

```
[root@vpn1 ~]# systemctl restart ipsec.service
```

```
[root@vpn1 ~]# ipsec auto --up net-to-net
```

VPN2

```
[root@vpn2 ~]# systemctl restart ipsec.service
```

```
[root@vpn2 ~]# ipsec auto --up net-to-net
```

必须两台都执行，否则不能成功。

显示 IPsec SA established tunnel mode 表示连接成功

```
122.100.2.11 ssh vpn1 ~$
[root@vpn1 ~]# systemctl restart ipsec.service
[root@vpn1 ~]# ipsec auto --up net-to-net
002 "net-to-net" #1: initiating main mode
104 "net-to-net" #1: STATE_MAIN_I1: initiate
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 500ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 1000ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 2000ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 4000ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 8000ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 16000ms for response
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 32000ms for response
001 "net-to-net" #1: max number of retransmissions (8) reached STATE_MAIN_I1. No response (or no acceptable response) to our first IKEv1 message
000 "net-to-net" #1: starting keying attempt 2 of an unlimited number, but releasing whack
[root@vpn1 ~]#
```

```
122.100.2.11 ssh vpn2 ~$
[root@vpn2 ~]# systemctl restart ipsec.service
[root@vpn2 ~]# ipsec auto --up net-to-net
002 "net-to-net" #1: initiating main mode
104 "net-to-net" #1: STATE_MAIN_I1: initiate
002 "net-to-net" #1: WARNING: connection net-to-net PSK length of 3 bytes is too short for sha2_256 PRF in FIPS mode (16 bytes required)
002 "net-to-net" #1: transition from state STATE_MAIN_I1 to state STATE_MAIN_I2
106 "net-to-net" #1: STATE_MAIN_I2: sent MI2, expecting MR2
002 "net-to-net" #1: transition from state STATE_MAIN_I2 to state STATE_MAIN_I3
108 "net-to-net" #1: STATE_MAIN_I3: sent MI3, expecting MR3
002 "net-to-net" #1: Main mode peer ID is ID.FQDN: 'vpn1'
002 "net-to-net" #1: transition from state STATE_MAIN_I3 to state STATE_MAIN_I4
004 "net-to-net" #1: STATE_MAIN_I4: ISAKMP SA established {auth=PRFRESHED_KEY cipher=aes_256 integ=sha2_256 group=MODP2048}
002 "net-to-net" #2: initiating Quick Mode PSK+ENCRI/PT+TUNNEL+PFS+UP+IKEV1_ALLOW+IKEV2_ALLOW+SAREF_TRACK+IKE_FRAG_ALLOW+ESN_NO {using isakmp#1 msgid=9cf18def propo
pfsgroup=MODP2048}
117 "net-to-net" #2: STATE_QUICK_I1: initiate
002 "net-to-net" #2: transition from state STATE_QUICK_I1 to state STATE_QUICK_I2
004 "net-to-net" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode {ESP=>0x6541b7dd <0xbdb2cddb xfrm=AES_256-HMAC_SHA2_256 NATOA=none NATD=none DPD=pa
[root@vpn2 ~]#
```

- 测试是否可用 由于只有两台机器，我们搭建虚拟内网网络来测试。

在 VPN1 上搭建虚拟网络 10.0.0.1/24 （步骤了解即可）

```
[root@vpn1 ~]# ip link add left1 type veth peer name left2
```

```
[root@vpn1 ~]# ip netns add left
```

```
[root@vpn1 ~]# ip link set left1 netns left
```

```
[root@vpn1 ~]# ip link set left2 up
```

```
[root@vpn1 ~]# ip addr add dev left2 10.0.0.1/24
```

```
[root@vpn1 ~]# ip netns exec left ip link set lo up
```

```
[root@vpn1 ~]# ip netns exec left ip link set left1 up
```

```
[root@vpn1 ~]# ip netns exec left ip addr add dev left1 10.0.0.2/24
```

[root@vpn1 ~]# ip netns exec left ip route add default via 10.0.0.1 `` 查看虚拟网络，可知绑定 IP 为 10.0.0.2

```
[root@vpn1 ~]# ip netns exec left ip a
```

```

[root@vpn1 ~]# ip link add left1 type veth peer name left2
[root@vpn1 ~]# ip netns add left
[root@vpn1 ~]# ip link set left1 netns left
[root@vpn1 ~]# ip link set left2 up
[root@vpn1 ~]# ip addr add dev left2 10.0.0.1/24
[root@vpn1 ~]# ip netns exec left ip link set lo up
[root@vpn1 ~]# ip netns exec left ip link set left1 up
[root@vpn1 ~]# ip netns exec left ip addr add dev left1 10.0.0.2/24
[root@vpn1 ~]# ip netns exec left ip route add default via 10.0.0.1
[root@vpn1 ~]# ip netns exec left ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
9: left1@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether ee:ah:5e:c0:63:4e brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.0.2/24 scope global left1
        valid_lft forever preferred_lft forever
    inet6 fe80::ecab:5eff:fec0:634e/64 scope link
        valid_lft forever preferred_lft forever
[root@vpn1 ~]#

```

在 VPN2 上搭建虚拟网络 10.0.1.1/24 （步骤了解即可）

```

[root@vpn2 ~]# ip link add left1 type veth peer name left2
[root@vpn2 ~]# ip netns add left
[root@vpn2 ~]# ip link set left1 netns left
[root@vpn2 ~]# ip link set left2 up
[root@vpn2 ~]# ip addr add dev left2 10.0.1.1/24
[root@vpn2 ~]# ip netns exec left ip link set lo up
[root@vpn2 ~]# ip netns exec left ip link set left1 up
[root@vpn2 ~]# ip netns exec left ip addr add dev left1 10.0.1.2/24
[root@vpn2 ~]# ip netns exec left ip route add default via 10.0.1.1
查看虚拟网络，可知绑定 IP 为 10.0.1.2
[root@vpn2 ~]# ip netns exec left ip a

```

```

[root@vpn2 ~]# ip link add left1 type veth peer name left2
[root@vpn2 ~]# ip netns add left
[root@vpn2 ~]# ip link set left1 netns left
[root@vpn2 ~]# ip link set left2 up
[root@vpn2 ~]# ip addr add dev left2 10.0.1.1/24
[root@vpn2 ~]# ip netns exec left ip link set lo up
[root@vpn2 ~]# ip netns exec left ip link set left1 up
[root@vpn2 ~]# ip netns exec left ip addr add dev left1 10.0.1.2/24
[root@vpn2 ~]# ip netns exec left ip route add default via 10.0.1.1
[root@vpn2 ~]# ip netns exec left ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN qlen 1
    link/gre 0.0.0.0 brd 0.0.0.0
3: gretap@NONE: <BROADCAST,MULTICAST> mtu 1462 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
4: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
9: left1@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether e2:63:71:7b:e5:76 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.1.2/24 scope global left1
        valid_lft forever preferred_lft forever
    inet6 fe80::e063:71ff:fe7b:e576/64 scope link
        valid_lft forever preferred_lft forever
[root@vpn2 ~]#

```

在 VPN1 上 PING 测试，可见可以 ping 通

```
[root@vpn1 ~]# ip netns exec left ping 10.0.1.2
```

```
[root@vpn1 ~]# ip netns exec left ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2): 56(84) bytes of data:
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=1.79 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.968 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=62 time=0.955 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=62 time=0.875 ms
64 bytes from 10.0.1.2: icmp_seq=5 ttl=62 time=0.794 ms
^C
--- 10.0.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.794/1.077/1.797/0.367 ms
[root@vpn1 ~]#
```

4. 基于 RSA Signature 认证方式(RSA 数字签名) 上面的认证方式是基于密码，现对不安全，现介绍如何使用数字签名模式认证

- 在 VPN1 和 VPN2 上分别生成一个新的 RSA 密钥对，记住后面的 key，后面会用到

```
[root@vpn1 ~]# rm -f /dev/random
```

```
[root@vpn1 ~]# ln -s /dev/urandom /dev/random
```

```
[root@vpn1 ~]# ipsec newhostkey --output /etc/ipsec.secrets
```

```
[root@vpn1 ~]# ipsec showhostkey --left --ckaid 对应的数
```

```
[root@vpn1 ~]# rm -f /dev/random
[root@vpn1 ~]# ln -s /dev/urandom /dev/random
[root@vpn1 ~]# ipsec newhostkey --output /etc/ipsec.secrets
/usr/libexec/ipsec/newhostkey: WARNING: "/etc/ipsec.secrets" exists, appending to it
Generated RSA key pair with CKaID 58837471fcb5e324d7fd8dad0c89787a498d52ff was stored in the NSS database
[root@vpn1 ~]# ipsec showhostkey --left --ckaid 58837471fcb5e324d7fd8dad0c89787a498d52ff
# rsakey 40EAC0B5
leftsecretkey=04AwEACns1dYy01ckvvh0AaC1FALqBpXOCVnmqrfLE0Kd0u6f0w4N8BLkyEpw04vBm5i6TrpA6SE/ECw0rc0E0B0uicqcyEF06vrtBsy6p1f69n4H3q5Zr3mje3D01Q12viJcA1LPFGvU0K2VABhY1Ybck62vYd1
z9L6MTqk3LctD0mHFD81qD2v44j9xj+HICj18QqXx0kVpVLEdULvndfs1T31WXP2AB/n4WPK3XdhF2LwJalLn1QWjvU2bp1QqWU1mneySvI0n1vZq320W0bLn0HbWNRULhcW7Zv81yGKFO/Q0/HTHIn1RR1Fq0+2THW1Q1nLzAand5A0rZcefSD
x3h0Q0hYK/ct3Q01bMyGd1Jkg+GdvPkXW0PLX1U0h0Hb0mA8AndDroTV08KnXndzMLNvZ3cHMSXXXP5ao/xBq3bzpNKJRA04wJ7ShqqWcZ5nQ044obKPSB1pvkx9av1bb07ThqQ081ZE1sM2MPL752NELJ24v0U9f98b17qBMLqpsJyE11w0Q02
zcQ0qK+Z44v1217080Vihqun1ZPFS0R0KUKIvILN0qHvP1p5vA1sQEqK3v2v1D32FAY1UDRno14tv4jFM0S46B50ewntKqpeH4QCC0QKZ3e0kzhZa0806v7B8K8M3pVugfh7s=
[root@vpn1 ~]#
```

VPN2 如下

```
[root@vpn2 ~]# rm -f /dev/random
```

```
[root@vpn2 ~]# ln -s /dev/urandom /dev/random
```

```
[root@vpn2 ~]# ipsec newhostkey --output /etc/ipsec.secrets
```

```
[root@vpn2 ~]# ipsec showhostkey --left --ckaid 对应的数
```

```
[root@vpn2 ~]# rm -f /dev/random
[root@vpn2 ~]# ln -s /dev/urandom /dev/random
[root@vpn2 ~]# ipsec newhostkey --output /etc/ipsec.secrets
/usr/libexec/ipsec/newhostkey: WARNING: "/etc/ipsec.secrets" exists, appending to it
Generated RSA key pair with CKaID 2edcfc62af1e38714961484305ca178454b4d696 was stored in the NSS database
[root@vpn2 ~]# ipsec showhostkey --left --ckaid 2edcfc62af1e38714961484305ca178454b4d696
# rsakey 40EACns1
leftsecretkey=04AwEACns1dYy01ckvvh0AaC1FALqBpXOCVnmqrfLE0Kd0u6f0w4N8BLkyEpw04vBm5i6TrpA6SE/ECw0rc0E0B0uicqcyEF06vrtBsy6p1f69n4H3q5Zr3mje3D01Q12viJcA1LPFGvU0K2VABhY1Ybck62vYd1
Yd8EvSLA081B0Q08v0v02PFP7p1z7L82SkjgRq58RA12fAtpv4uvq540+U8t4vhl/zsyh27E1prncC1Se13tth0K9h/19tSuBk08B0W0qtu17WNS288Fck1030FhSNY0Pkcq5qZe5Z2LEBhtvQD0MDL1ET11PRY30eWdUv1pvYK310myNmd/j
62x9CvM6/EBHpkpk0e413142m3q39F064Hgy0Fvqv02weasye/0akK9xrBQW9Y0fKH00G0u2C0Jy2fx4ANFYrdP7IusF3FLMwXDR1ShcK1Ld88v50kqge0=
[root@vpn2 ~]#
```

- 修改 VPN1 和 VPN2 的配置文件如下图（两机器配置文件相同）

```
vim /etc/ipsec.conf
```

conn net-to-net

```
# 一端 IP 地址
```

```
left=192.168.1.11
```

```
# 一端内网网段地址
```

```
leftsubnet=10.0.0.0/24
```

```
# 一端的标识符，可以任意填写，如果多个连接需要区分
```

```
leftid=@vpn1
```

```
leftnexthop=%defaulttroute
```


leftsasigkey= (VPN1 下的数)

right=192.168.2.11

rightsubnet=10.0.1.0/24

rightid=@vpn2

rightnexthop=%defaultroute

rightrsasigkey= (VPN2 下的数)

add 代表只是添加, 但并不会连接, 如果为 start 则代表着启动自动连接

auto=add

```
# It is best to add your IPsec connections as separate files in /etc/ipsec.d/
include /etc/ipsec.d/*.conf

conn net-to-net
    left=192.168.1.11
    leftsubnet=10.0.0.0/24
    leftid=@vpn1
    leftnexthop=%defaultroute
    leftsasigkey=0aAeAAd8573563Q2qKRYdt5fW7dZ/0qwk0kvt02v0MM/U6hAu0u5Rq13VuJp2v8Rv3V52svf12gn0Bcuz3LEHFSmPhfdU0b2fpx1h4qfL1fgig8q18ntNc9tJo7d3bxUOfh1DoBLrXQ1zSLQcTBM2No0hJ5Co/
    zBL6N1qK3LctD0eHJfD81qD2Y44k9xj+hHicj9Qx9XoXkVpVXLdULndfs1T3WRXP2AB/n4h4PK3dHF2LwJsu10WjvU2bp1DgNu1an+ySV1Gn1v3q32MOBlnDhWNRuLhCwF7z6BlyGKFO/06/hTHiW1FR1FqX0x2THW1Q1nL2Aand5AdZcef50
    x3Hn008nyK/cTp3QIDMyjGDI1Kq3dVFK8XwP9XLU1oHohb0mAS4ndrotvD8KXidMLNvZ3CHSXXXP8ao/xBqJ0xpkKJRA04w/7ShqqKcZ5n0Q440bKPSB1pvkx9w1b6g7THqQcB1ZELsK2MPL75ZNL1Z4w0U9f96b17qBM1qpsjyEt1xW0ge2
    zcQqKwZE1v17588v1HqnuJ2P5P5oQkUpInL0g9vlp5wA1sqBqK3v2D52FA1UDPno14tv43HKS4B55eWnTKpeH4C00QqJ23e0kzhZaD809tBqKdCpVugH77s=

    right=192.168.2.11
    rightsubnet=10.0.1.0/24
    rightid=@vpn2
    rightnexthop=%defaultroute
    rightrsasigkey=0aAeAAd8573563Q2qKRYdt5fW7dZ/0qwk0kvt02v0MM/U6hAu0u5Rq13VuJp2v8Rv3V52svf12gn0Bcuz3LEHFSmPhfdU0b2fpx1h4qfL1fgig8q18ntNc9tJo7d3bxUOfh1DoBLrXQ1zSLQcTBM2No0hJ5Co/
    1YdE5vSwD8e10H50MMHfK4n7AEKvK9Mc0h0n0LgPhgXFPATLXY40V6zD8P02K6vZuR1eNMTMxVmmprCOfacB7BmK15d0mFBUK2vue20uA1k135V1MCdp/83PALVeWMLdtDY6Hf+11RMSdCtU0128868JN2FkeZE/08Xutv1a/JMAU1yak00
    /PstufRy44EEp02v288v02P70rZULZ5K1jgRkzSBRA1z7A1p4uq54Q4u9R14wLZsyhZ2ElprgC118uG3u8kSh19tSu8B8S0QW0tu17WV528P8KtU3UfH5NWPkCq5GZe5ZUebMt+Q0MDUJtEF11PKY300eNdw1pvYK31DMYjnd/
    16ZQVME/HEH7PHK441314FwS2S8P804Hgy0FTvgvQ2wasse/0akX9r8QW97GfK5002u0200j2FzA4MPrF77usFJdFLUwXDR1Shc1Ld88v50kg8e0=

    auto=add

/etc/ipsec.conf* 66L, 3647C
```

• 重新启动服务

VPN1

[root@vpn1 ~]# systemctl restart ipsec.service

[root@vpn1 ~]# ipsec auto --up net-to-net

VPN2

[root@vpn2 ~]# systemctl restart ipsec.service

[root@vpn2 ~]# ipsec auto --up net-to-net

必须两台都执行, 否则不能成功。

显示 IPsec SA established tunnel mode 表示连接成功

```
[root@vpn1 ~]# systemctl restart ipsec.service
[root@vpn1 ~]# ipsec auto --up net-to-net
002 "net-to-net" #1: initiating Main Mode
104 "net-to-net" #1: STATE_MAIN_I1: initiate
002 "net-to-net" #1: transition from state STATE_MAIN_I1 to state STATE_MAIN_I2
106 "net-to-net" #1: STATE_MAIN_I2: sent MI2, expecting MR2
002 "net-to-net" #1: transition from state STATE_MAIN_I2 to state STATE_MAIN_I3
108 "net-to-net" #1: STATE_MAIN_I3: sent MI3, expecting MR3
002 "net-to-net" #1: Main mode peer ID is ID_IPsec @vpn2
002 "net-to-net" #1: transition from state STATE_MAIN_I3 to state STATE_MAIN_I4
004 "net-to-net" #1: STATE_MAIN_I4: ISAKMP SA established (auth=RSA SIG cipher=aes_256 integ=sha group=MODP2048)
002 "net-to-net" #2: initiating Quick Mode
117 "net-to-net" #2: STATE_QUICK_I1: initiate
002 "net-to-net" #2: transition from state STATE_QUICK_I1 to state STATE_QUICK_I2
004 "net-to-net" #2: STATE_QUICK_I2: sent QI2, [IPsec SA established tunnel] mode {ESP=>0x89aff838 <0x061001ff xfrm=AES_128-HMAC_SHA1 NAT0=none NATD=none DPD=passive}

[root@vpn2 ~]# systemctl restart ipsec.service
[root@vpn2 ~]# ipsec auto --up net-to-net
002 "net-to-net" #3: initiating Quick Mode
117 "net-to-net" #3: STATE_QUICK_I1: initiate
002 "net-to-net" #3: transition from state STATE_QUICK_I1 to state STATE_QUICK_I2
004 "net-to-net" #3: STATE_QUICK_I2: sent QI2, [IPsec SA established tunnel] mode {ESP=>0xd2e65b23 <0x62940f0b xfrm=AES_128-HMAC_SHA1 NAT0=none NATD=none DPD=passive}

[root@vpn2 ~]# cat /etc/ipsec.conf
```

• 检测

在 VPN1 上 PING 测试, 可见可以 ping 通

[root@vpn1 ~]# ip netns exec left ping 10.0.1.2

```
[root@vpn1 ~]# ip netns exec left ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data:
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=1.74 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=1.10 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=62 time=0.945 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=62 time=1.08 ms
^C
--- 10.0.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.945/1.220/1.747/0.311 ms
[root@vpn1 ~]#
```

2.5 任务五

2.5.1 任务描述

本实验任务基于真实企业网络环境，在两台服务器搭建的典型企业局域网环境中，主要完成以下内容：

- （1）搭建 overlay 网络实现不同宿主机之间同网段机器相通。
- （2）检测网络联通性。

2.5.2 实验目标

- 了解 overlay 网络的使用场景。
- 掌握 overlay 搭建和使用。
- 掌握 openvswitch 的使用。

2.5.3 实验工具

- openvswitch

2.5.4 操作步骤

1. 在 VPN1 和 VPN2 分别安装 openvswitch 并启动服务

安装 openvswitch

```
[root@vpn1 ~]# yum install openvswitch -y
```

启动服务

```
[root@vpn1 ~]# systemctl start openvswitch.service
```

查看服务状态

```
[root@vpn1 ~]# systemctl status openvswitch.service
```

```
[root@vpn1 ~]# yum install openvswitch -y
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvswitch.x86_64.0.2.5.0-2.el7 将被 安装
--> 解决依赖关系完成
```

依赖关系解决

```
=====
Package                                架构
=====
正在安装：
openvswitch                            x86_64
```

事务概要

```
[root@vpn1 ~]# systemctl start openvswitch.service
[root@vpn1 ~]# systemctl status openvswitch.service
● openvswitch.service - Open vSwitch
   Loaded: loaded (/usr/lib/systemd/system/openvswitch.service; disabled; vendor preset: disabled)
   Active: active (exited) since 三 2024-04-24 20:59:09 CST; 11s ago
   Process: 13993 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 13993 (code=exited, status=0/SUCCESS)

4月 24 20:59:09 vpn1 systemd[1]: Starting Open vSwitch...
4月 24 20:59:09 vpn1 systemd[1]: Started Open vSwitch.
[root@vpn1 ~]#
```

```
[root@vpn2 ~]# yum install openvswitch -y
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvswitch.x86_64.0.2.5.0-2.el7 将被 安装
--> 解决依赖关系完成
```

依赖关系解决

```
=====
Package                                架构                                版本
=====
正在安装:
openvswitch                            x86_64                             2.5
=====
事务概要
=====
安装 1 软件包
```

```
[root@vpn2 ~]# systemctl start openvswitch.service
[root@vpn2 ~]# systemctl status openvswitch.service
● openvswitch.service - Open vSwitch
   Loaded: loaded (/usr/lib/systemd/system/openvswitch.service; disabled; vendor preset: disabled)
   Active: active (exited) since 三 2024-04-24 21:00:23 CST; 8s ago
   Process: 13905 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 13905 (code=exited, status=0/SUCCESS)

4月 24 21:00:23 vpn2 systemd[1]: Starting Open vSwitch...
4月 24 21:00:23 vpn2 systemd[1]: Started Open vSwitch.
[root@vpn2 ~]#
```

2. 配置 VPN1

在 VPN1 上添加名为 br0 的网桥:

```
ovs-vsctl add-br br0
```

给 br0 网桥分配一个 ip

```
ifconfig br0 10.1.0.1/24 up
```

```
[root@vpn1 ~]# ovs-vsctl add-br br0
[root@vpn1 ~]# ifconfig br0 10.1.0.1/24 up
[root@vpn1 ~]# ifconfig br0
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.0.1 netmask 255.255.255.0 broadcast 10.1.0.255
    inet6 fe80::243e:b3ff:fedb:ff42 prefixlen 64 scopeid 0x20<link>
    ether 26:3e:b3:db:ff:42 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 648 (648.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. 配置 VPN2

在 VPN1 上添加名为 br0 的网桥:

```
ovs-vsctl add-br br0
```

给 br0 网桥分配一个 ip

```
ifconfig br0 10.1.0.2/24 up
```

```
[root@vpn2 ~]# ovs-vsctl add-br br0
[root@vpn2 ~]# ifconfig br0 10.1.0.2/24 up
[root@vpn2 ~]# ifconfig br0
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.0.2 netmask 255.255.255.0 broadcast 10.1.0.255
    inet6 fe80::90e4:2cff:fe17:ce47 prefixlen 64 scopeid 0x20<link>
    ether 92:e4:2c:17:ce:47 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6 bytes 508 (508.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@vpn2 ~]#
```

4. 搭建 VXLAN 隧道

在 VPN1 上设置 VXLAN，远端 ip 设置为 VPN2 能对外通信的 br0 的 ip。
 ovs-vsctl add-port br0 vx1 -- set interface vx1 type=vxlan options:remote_ip=192

.168.2.11

查看

ovs-vsctl show

```
[root@vpn1 ~]# ovs-vsctl add-port br0 vx1 -- set interface vx1 type=vxlan options:remote_ip=192.168.2.11
[root@vpn1 ~]# ovs-vsctl show
4929dfe8-830c-4db7-8d3c-e2431ce8dc0c
Bridge "br0"
  Port "br0"
    Interface "br0"
      type: internal
  Port "vx1"
    Interface "vx1"
      type: vxlan
      options: {remote_ip="192.168.2.11"}
ovs_version: "2.5.0"
[root@vpn1 ~]#
```

• 在 Host2 上设置 VXLAN，远端 ip 设置为 Host1 能对外通信的 br0 的 ip。

ovs-vsctl add-port br0 vx1 -- set interface vx1 type=vxlan options:remote_ip=192

.168.1.11

查看

ovs-vsctl show

```
[root@vpn2 ~]# ovs-vsctl add-port br0 vx1 -- set interface vx1 type=vxlan options:remote_ip=192.168.1.11
[root@vpn2 ~]# ovs-vsctl show
cebea76f-7c41-4573-ba0a-fe4bb0ed4b43
Bridge "br0"
  Port "vx1"
    Interface "vx1"
      type: vxlan
      options: {remote_ip="192.168.1.11"}
  Port "br0"
    Interface "br0"
      type: internal
ovs_version: "2.5.0"
[root@vpn2 ~]#
```

5. 验证 VxLAN 隧道在 VPN1 上 ping 10.1.0.2 发现可以通

```
ovs_version: "2.5.0"
[root@vpn1 ~]# ping 10.1.0.2
PING 10.1.0.2 (10.1.0.2) 56(84) bytes of data:
 64 bytes from 10.1.0.2: icmp_seq=1 ttl=64 time=2.23 ms
 64 bytes from 10.1.0.2: icmp_seq=2 ttl=64 time=0.945 ms
 64 bytes from 10.1.0.2: icmp_seq=3 ttl=64 time=0.939 ms
 64 bytes from 10.1.0.2: icmp_seq=4 ttl=64 time=0.844 ms
^C
--- 10.1.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.844/1.239/2.230/0.574 ms
[root@vpn1 ~]#
```

三、实验总结

在这次 VPN 实验中，我深入了解并亲手实践了从基础的 IP 隧道搭建到复杂的 OpenVPN 和 IPsec 配置的多种 VPN 技术。

首先，通过配置 IP 隧道和 GRE 技术，成功实现了两个不同网络之间的连接。这个过程让我深刻理解了数据包在网络中的封装和传输机制，尤其是如何通过隧道技术跨越网络隔离。

接着，通过使用 OpenSSL 来生成和管理密钥，我进一步掌握了公钥和私钥的应用，这对于任何涉及加密的技术都是基础且必须的。

在搭建 OpenVPN 和配置 IPsec VPN 的过程中，我遇到了配置复杂性和故障排除的挑战。每一步都需要精确的配置和对安全需求的严格理解，任何小小的疏漏都可能导致通信失败或安全漏洞。

Overlay 网络的搭建则是我在这次实验中接触的高级部分，它涉及到了现代云计算中广泛应用的虚拟化技术。通过配置 Openvswitch，我不仅学到了如何在物理网络之上构建虚拟网络，还实际体验了这种技术在实现跨主机通信中的强大功能。