**第 五 次 实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | **网络安全实验** | | | | |
| 学生姓名 | 应晓宇 | 学号 | 2021302181140 | 指导老师 | 陈治宏 |
| 专业 | 网安 | 班级 | 7 | 实验时间 | 24.4.24 |

1. **实验介绍**
   * + 1. **实验名称：VPN实验**
       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. **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. **实验工具：**

**IP**

**Openssl**

**Ipsec**

**ovs-vsctl**

**openvpn**

**tcpdump**

**sysctl**

**modprobe**

**iptables**

* + - 1. **实验环境：**



1. **实验内容**
2. 任务一

【任务描述】

本实验任务基于真实企业网络环境，在两台不同网络环境的环境中，主要完成以下内容：

实现两不同网络内的内网通过ip隧道使之互通并检测。

【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。

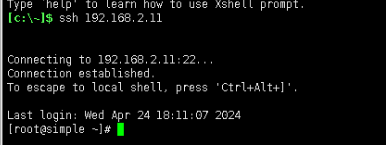
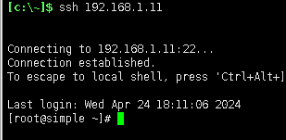
【实验工具】

ip

modprobe

【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



分别修改主机名：

# hostnamectl set-hostname vpn1

# hostnamectl set-hostname vpn2



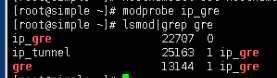
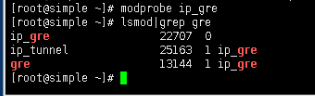
2.vpn1和vpn2主机分别加载gre内核模块并检查

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。

3.1 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



3.2启动gre1并分配ip地址10.10.10.1，检测是否添加并启动。

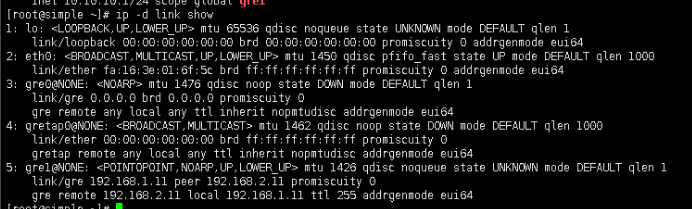
# ip link set gre1 up

# ip addr add 10.10.10.1/24 dev gre1

# ip a | grep gre1



3.3 查看隧道状态



3.3 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。

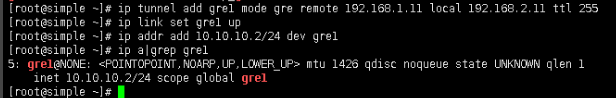
操作步骤如下

# ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11 ttl 255

# ip link set gre1 up

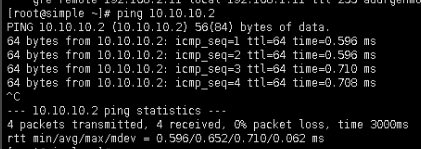
# ip addr add 10.10.10.2/24 dev gre1

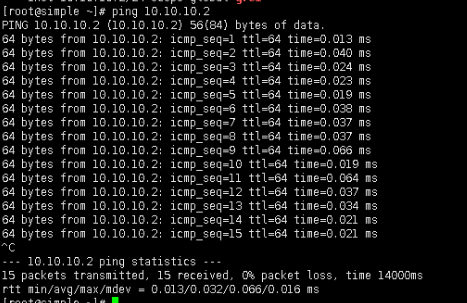
# ip a | grep gre1



3.4测试隧道是否通

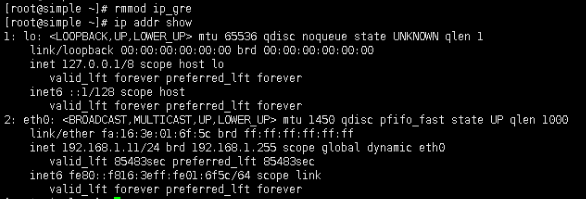
ping检测





4.卸载GRE模块

# rmmod ip\_gre



1. 任务二

【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秘钥证书管理、对称加密和非对称加密。

【实验目标】

1.了解openssl加密解密原理。

2.掌握openssl如何生成公钥私钥，以及公私钥之间的相互转化。

3.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。

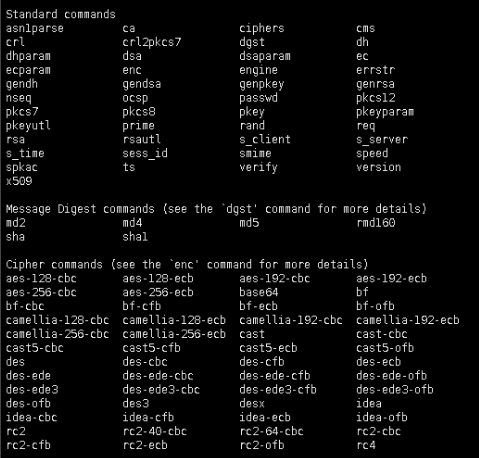
4.掌握如何生成带签名信息的证书。

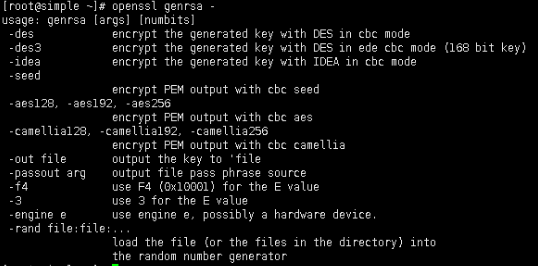
【实验工具】

openssl

【操作步骤】

1.查看openssl命令的基本帮助





2.生成私钥

2.1生产RSA私钥(无加密)

[root@vpn1 ~]# openssl genrsa -out rsa\_private.key 2048

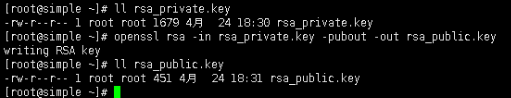
[root@vpn1 ~]# ll rsa\_private.key



2.2生成rsa\_private.key私钥对应的公钥

[root@vpn1 ~]# openssl rsa -in rsa\_private.key -pubout -out rsa\_public.key

[root@vpn1 ~]# ll rsa\_public.key



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\_ase\_public.key

[root@vpn1 ~]# ll rsa\_\*



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

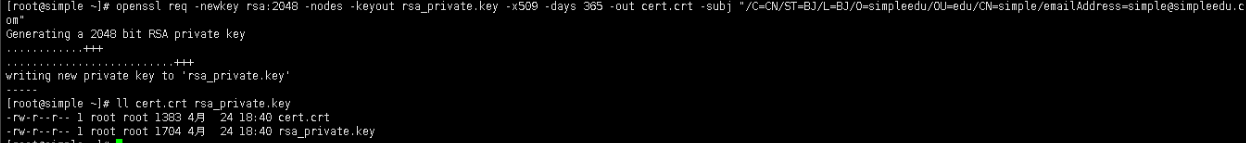


5.生成自签名证书

# 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



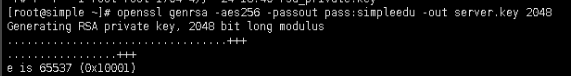
6.生成签名请求及CA 签名

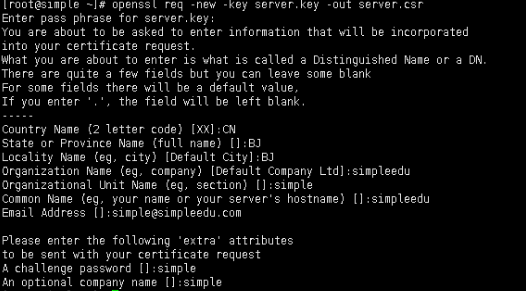
# 使用 RSA私钥生成 CSR 签名请求

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 \*





1. 任务三

【任务描述】

本实验任务基于真实企业网络环境，在两台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

【实验目标】

1.了解企业级别openvpn的使用场景。

2.掌握企业级别openvpn搭建和使用。

3.掌握openvpn客户端与服务端的搭建配置。

【实验工具】

openvpn

【操作步骤】  
1.在vpn1机器安装openvpn并验证

1.1 修改yum源

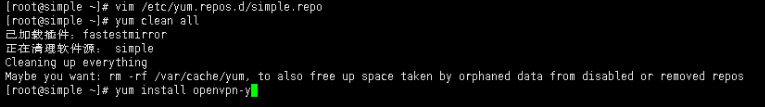
输入：vim /etc/yum.repos.d/simple.repo

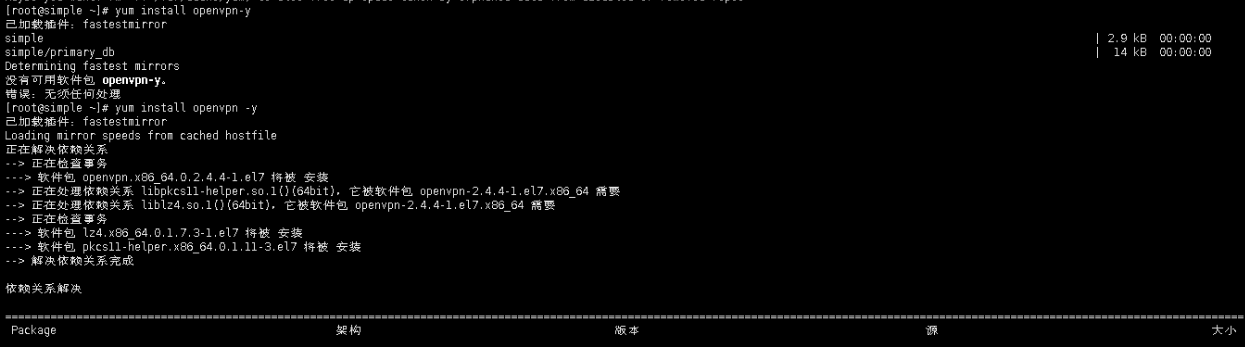
将 baseurl=http://192.168.50.50/yum 修改为 baseurl=file:///data/repos

i 进入插入状态，修改完成后输入 :wq 保存退出

[root@vpn1 ~]# yum clean all

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







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

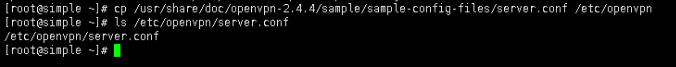


2.修改openvpn的配置文件server.conf配置文件的内容如下

2.1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



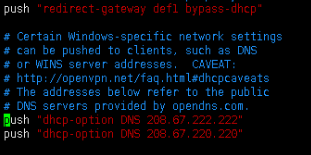
2.2 修改openvpn服务端的配置文件 /etc/openvpn/server.conf

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

2.2.1 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)



2.2.2打开这三行注释，配置DNS（实验环境无法连通外网，可不配置）



2.2.3 设置启动用户



2.2.4 注释掉 explicit-exit-notify 1



3.安装密钥生成软件

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



4.准备配置证书文件

4.1拷贝文件到/etc/openvpn

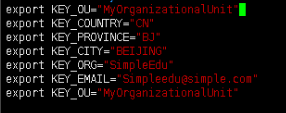
[root@vpn1 ~]# cp -r /usr/share/easy-rsa/ /etc/openvpn/

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



4.2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



使配置的环境变量生效



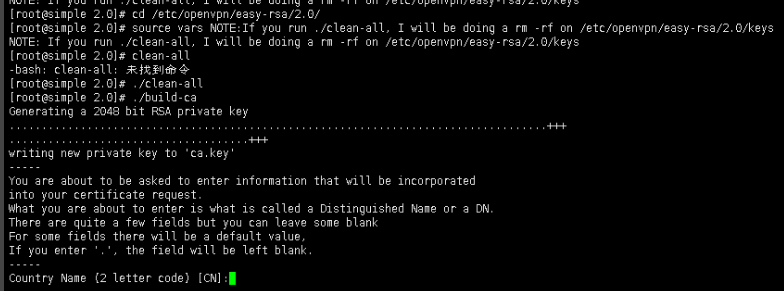
4.3 根据提示先删除所有，再根据自己情况进行修改

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

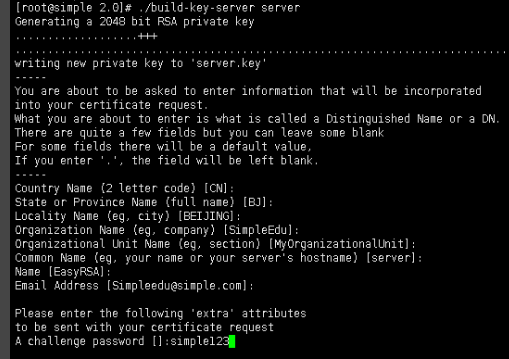
[root@vpn1 2.0]# source vars NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/2.0/keys

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

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



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



生成防攻击的key文件（防DDos攻击、UDP淹没等恶意攻击）

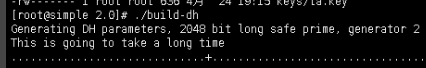
[root@vpn1 2.0]# openvpn --genkey --secret keys/ta.key

[root@vpn1 2.0]# ll keys/ta.key



1. 建客户端证书 6.1.创建密钥文件，耗时间一分钟左右

[root@vpn1 2.0]# ./build-dh



可以看到有一个dh2048.pem的文件产生



6.2拷贝密钥认证文件到配置文件目录下

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

[root@vpn1 keys]# cp dh2048.pem ca.crt server.crt server.key ta.key /etc/openvpn



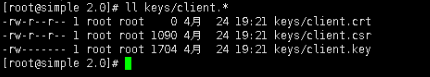
6.3创建一个通用名(common name)为 client的客户端证书，交互输入自己的值,默认回车键进行

[root@vpn1 keys]# cd ..

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

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





7.启动并检查 7.1 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

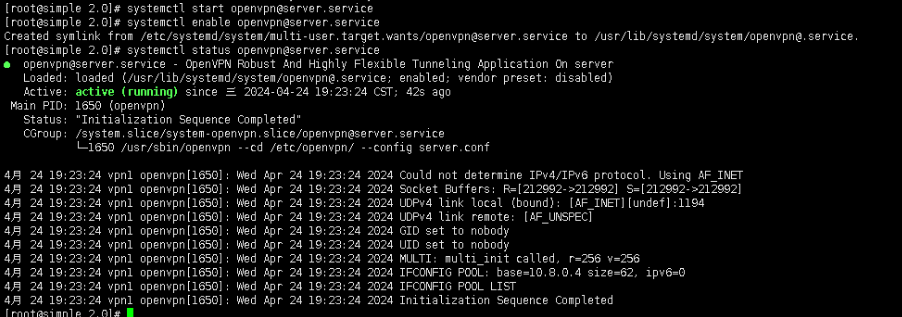
[root@vpn1 ~]# systemctl status openvpn@server.service

# 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

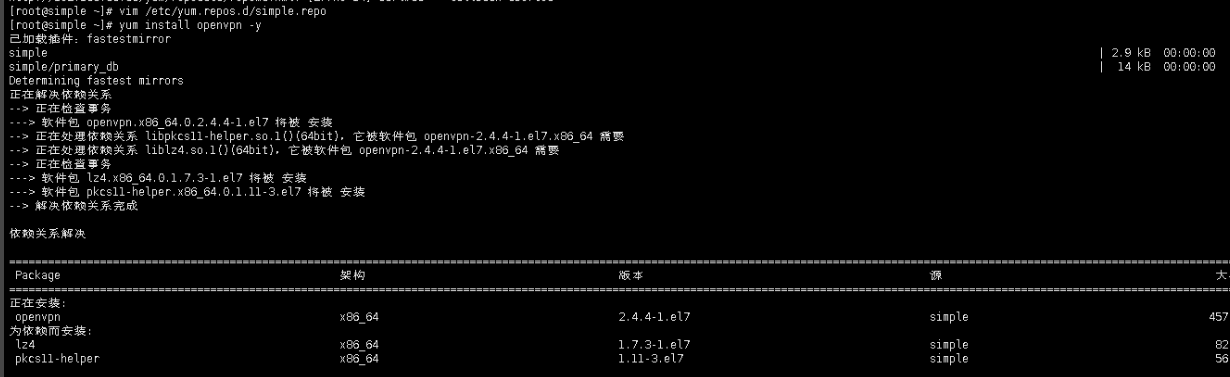




8.客户端（vpn2）登录测试

8.1 在客户端安装openvpn

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

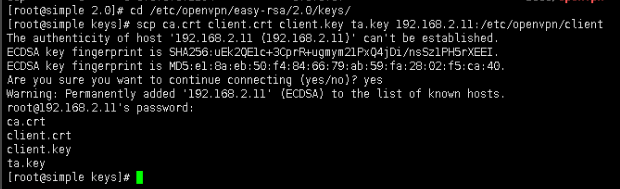


8.2 在vpn1端把生产文件拷贝到客户端

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

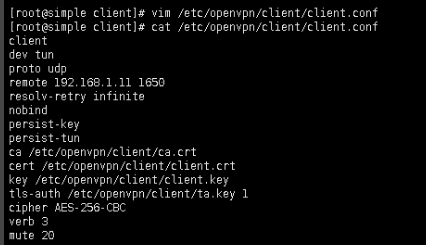
\# 密码为Simplexue123

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



8.3 编辑客户端配置文件

[root@vpn2 ~]# vim /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



8.4 启动openvpn客户端并挂后台运行，并可实时查看其日志。

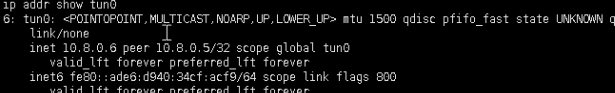
[root@vpn2 client]# cd /etc/openvpn/client/

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

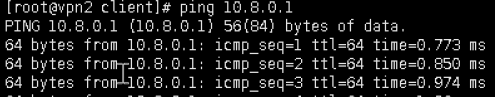


8.5 查看网卡信息，得知已获取到ip

[root@vpn2 ~]# ip addr show tun0



8.6 测试是否可使用 ```shell [root@vpn2 client]# ping 10.8.0.1 ```

S

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



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



8.8 关闭服务 ```shell [root@vpn1 ~]# pkill openvpn [root@vpn2 ~]# pkill openvpn ```



1. 任务四

【任务描述】

本实验任务基于真实企业网络环境，在两台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

【实验目标】

1.了解企业级别ipsec的使用场景。

2.掌握企业级别ipsec搭建和使用。

3.掌握ipsec客户端与服务端的搭建配置。

4.掌握ipsec多种验证方式的实现

【实验工具】

ipsec

openssl

【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

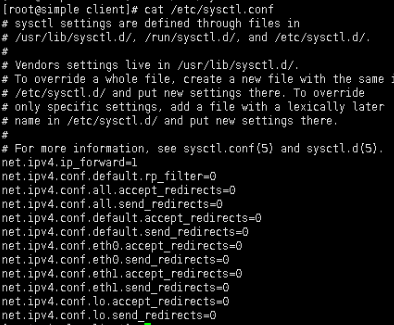
net.ipv4.conf.eth0.send\_redirects = 0

net.ipv4.conf.eth1.accept\_redirects = 0

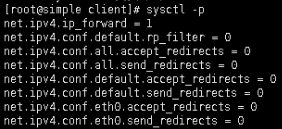
net.ipv4.conf.eth1.send\_redirects = 0

net.ipv4.conf.lo.accept\_redirects = 0

net.ipv4.conf.lo.send\_redirects = 0



# 使配置生效[root@vpn1 ~]# sysctl -p



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

2.1安装并验证，两台机器都做

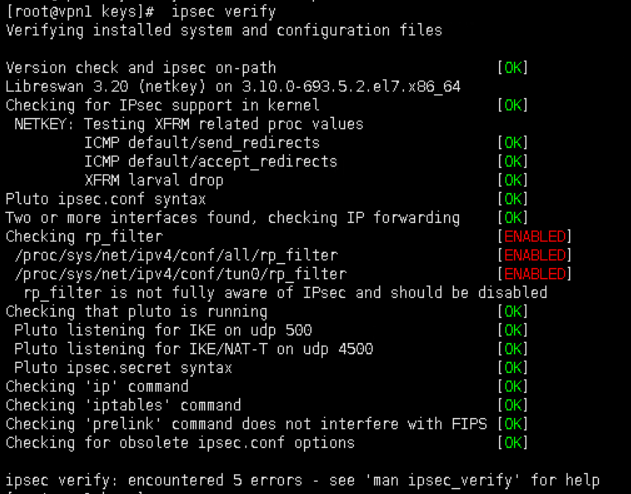
[root@vpn1 ~]# yum install openswan libreswan -y

[root@vpn1 ~]# ipsec --version

2.2 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 ~]# 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



3.配置ipsecVPN配置（模式为network-to-network）

3.1基于pre-shared keys认证方式（PSK） 配置/etc/ipsec.conf



3.2两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"



3.3两端重新启动服务，并验证。

VPN1

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

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

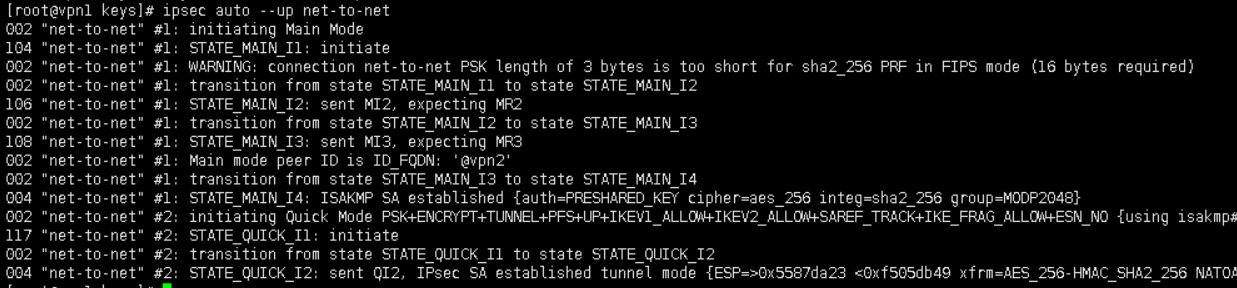
VPN2

[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



3.4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

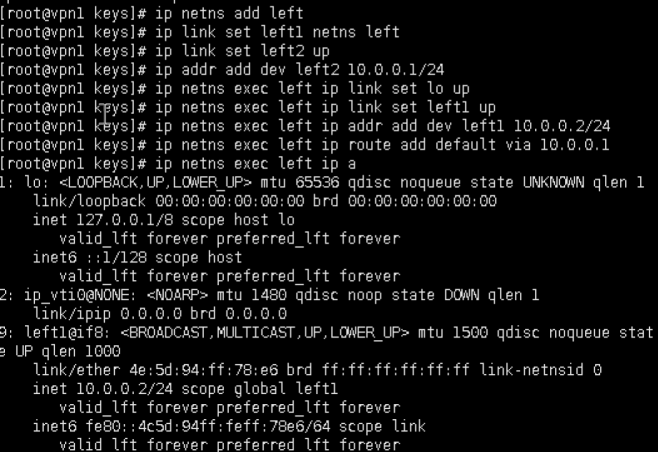
[root@vpn1 ~]# ip netns exec left ip link set lo up

[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



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

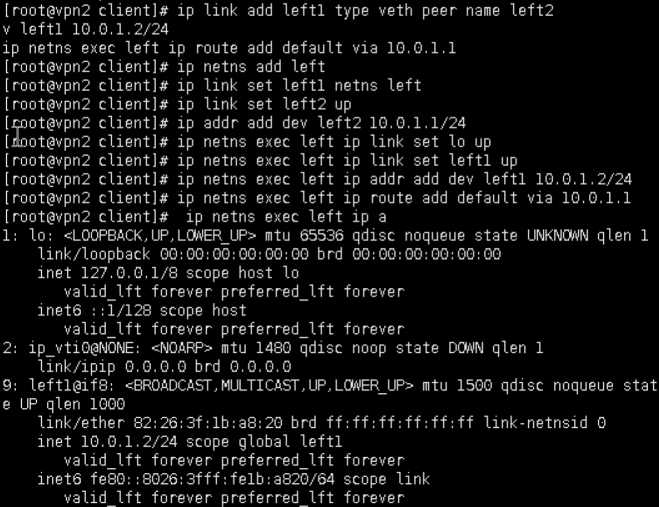
[root@vpn2 ~]# ip netns exec left ip link set lo up

[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

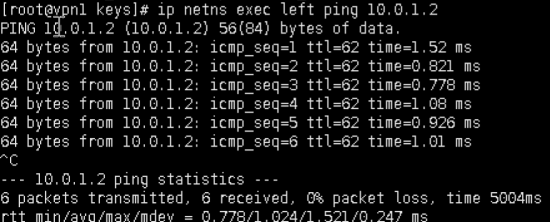
查看虚拟网络，可知绑定IP为10.0.1.2

[root@vpn2 ~]# ip netns exec left ip a



在VPN1上PING测试，可见可以ping通

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



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

4.1 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

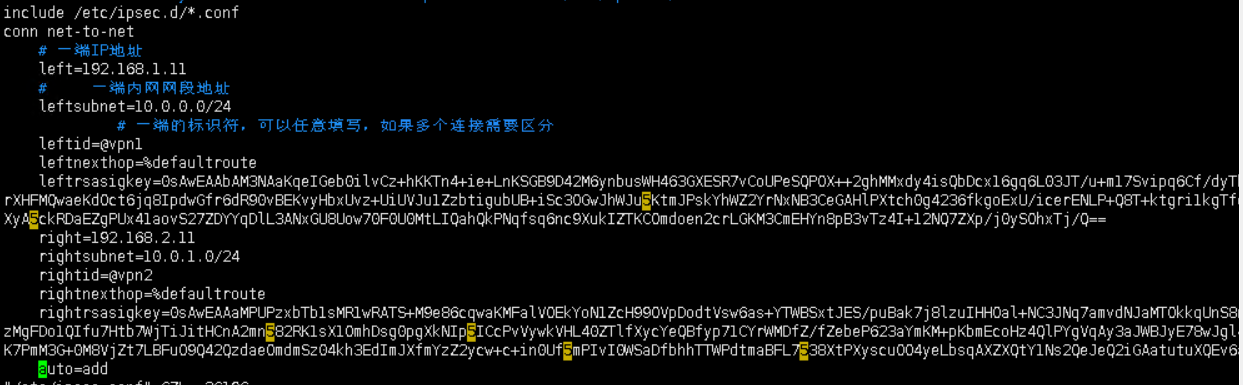
[root@vpn1 ~]# ln -s /dev/urandom /dev/random

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



[root@vpn1 ~]# ipsec showhostkey --left --ckaid

4.2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



4.3重新启动服务

VPN1

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

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

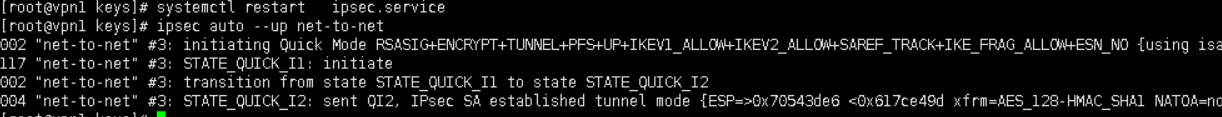
VPN2

[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。

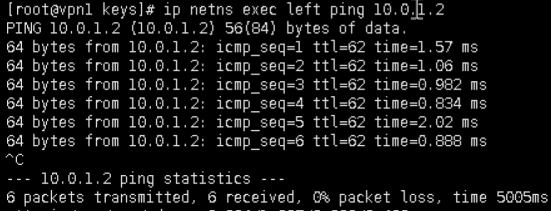
显示IPsec SA established tunnel mode 表示连接成功



4.4 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



1. 在VPN1和VPN2上清除虚拟内网，停止服务。两台机器都执行。

[root@vpn1 ~]# ip netns del left [root@vpn1 ~]# systemctl stop ipsec

1. 任务五

【任务描述】

本实验任务基于真实企业网络环境，在两台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。

【实验目标】

1.了解overlay网络的使用场景。

2.掌握overlay搭建和使用。

3.掌握openvswitch的使用。

【实验工具】

openvswitch

【操作步骤】

1.在VPN1和VPN2分别安装openvswitch并启动服务

# 安装openvswitch

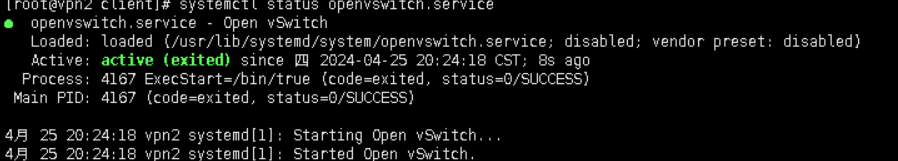
[root@vpn1 ~]# yum install openvswitch -y

# 启动服务

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

# 查看服务状态

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



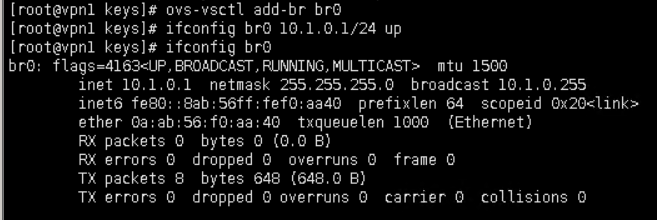
2.配置VPN1

# 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



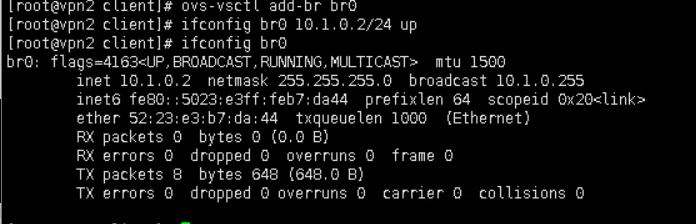
3.配置VPN2

# 在VPN2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



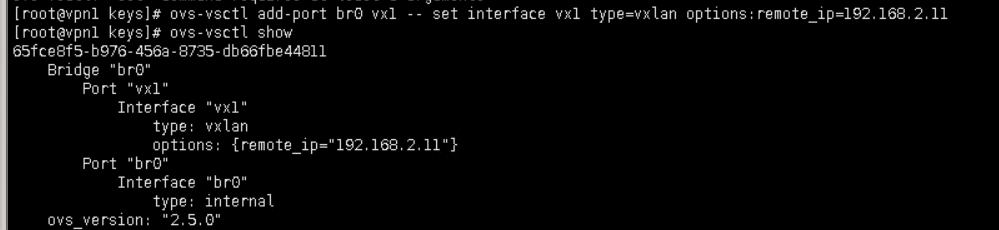
4.搭建VXLAN隧道

# 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

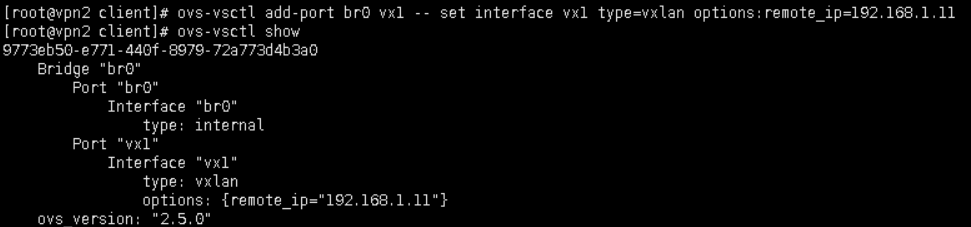
# 查看

ovs-vsctl show

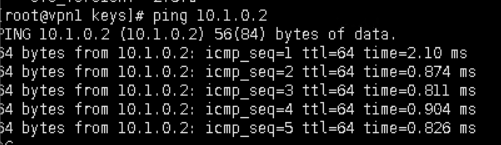


在VPN2上设置VXLAN，远端ip设置为Host1能对外通信的br0的ip。

ovs-vsctl add-port br0 vx1 -- set interface vx1 type=vxlan options:remote\_ip=192.168.1.11



5.验证VxLAN隧道 在VPN1上ping 10.1.0.2 发现可以通



三、实验总结

这几个任务涵盖了网络安全和虚拟网络的重要主题。通过完成这些任务，我获得了宝贵的经验和技能。

首先，使用IP命令搭建基于隧道的虚拟专有网络（VPN）是一项关键的网络技能。通过这个任务，我学会了如何使用IP命令配置和管理Linux系统中的网络隧道，实现安全的虚拟专有网络连接。这个过程提高了我对网络隧道和VPN原理的理解，并加强了我在Linux系统管理方面的能力。其次，使用OpenSSL创建加密密钥是网络安全中必不可少的一步。通过这个任务，我学会了使用OpenSSL工具生成和管理加密密钥，以确保数据的安全性和保密性。这个经验对我理解公钥基础设施（PKI）和加密算法有了更深入的了解。接着，学习了SSL VPN之OpenVPN的安装配置。OpenVPN是一个流行且强大的开源VPN解决方案，通过这个任务，我了解了如何在不同的操作系统上安装和配置OpenVPN，为远程访问和安全通信提供了重要的解决方案。然后，深入研究了IPsec VPN的原理和安装配置。IPsec是一种常见的VPN协议，通过这个任务，我学到了如何使用IPsec实现安全的站点到站点或远程访问连接。这个过程加深了我对IPsec协议套件及其工作原理的理解。最后，通过实现基于Overlay技术的隧道网络，我进一步认识了云计算中的网络虚拟化和Overlay技术。这个任务使我了解了如何在云环境中利用Overlay技术构建安全的隧道网络，实现多租户和多地域的网络连接。

总体而言，这些任务不仅扩展了我的网络知识和技能，还提升了我在网络安全和虚拟网络领域的实际操作能力。这些经验对我未来在网络和系统管理领域的职业发展具有重要意义。