operation 06

=============================================

**一，使用tomcat搭建https网站**

**1，准备工作，先还原环境**

killall java

cd ~/lnmp\_soft

rm -rf /usr/local/tomcat/

cp -r apache-tomcat-8.0.30 /usr/local/tomcat

yum -y install java-1.8.0-openjdk

cd /usr/local/tomcat/

bin/startup.sh //开启服务

netstat -ntulp | grep java //检查端口，需要看到8005、

8009、8080端口

vim conf/server.xml //修改配置文件，打开找到85行，将前后的注释删除 <!-- -->

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="/usr/local/tomcat/keystore"

keystorePass="123456" /> //这里

添加keystoreFile(指定公钥私钥文件)和keystorePass(公钥私钥文件密码)

keytool -genkeypair -alias tomcat -keyalg RSA

-keystore /usr/local/tomcat/keystore //创建密钥对，别名是tomcat，加

密算法是RSA，最后是存储位置

输入密钥库口令: 输入123456

再次输入新口令: 再次输入123456

您的名字与姓氏是什么?

[Unknown]: dc

。。。。。

CN=dc, OU=dc, O=dc, L=dc, ST=dc, C=dc是否正确? 下课 10:10回

[否]: y

输入 <tomcat> 的密钥口令

(如果和密钥库口令相同, 按回车): 直接回车

Warning:

ls //检查下/usr/local/tomcat下面会产生keystore则成功

bin/shutdown.sh //重启tomcat

bin/startup.sh

netstat -ntulp | grep java //检查可以看到8443端口

echo "tomcat-https~~" > webapps/ROOT/index.html //准备测试页面

curl -k <https://192.168.2.5:8443> //访问安全加密网站

------------------------------------------------------------

**二，将源码包转换为rpm包**

[root@proxy ~]# yum -y install rpm-build //安装制作rpm包的工具

[root@proxy ~]# rpmbuild -ba nginx.spec //制作rpm包，但是没有配

置文件会报错，报错也需要敲，会产生所需的目录

[root@proxy ~]# ls //可以看到一个rpmbuild目录

[root@proxy ~]# cp ~/lnmp\_soft/nginx-1.17.6.tar.gz rpmbuild/SOURCES/

//拷贝源码包到rpmbuild的工作目录

[root@proxy ~]# yum -y install gcc make pcre-devel openssl-devel

vim rpmbuild/SPECS/nginx.spec //编写将nginx转换为rpm包的配置文件

Name:nginx //软件名

Version:1.17.6 //版本

Release:1 //发布的rpm包的版本

Summary:test~ //简单描述

#Group:

License:GPL //授权协议 ，GPL表示自由软件

URL:www.abc.com //网址

Source0:nginx-1.17.6.tar.gz //源码包

#BuildRequires:

#Requires:

%description //详细描述

test~ test~

%post //可以添加安装rpm包之后执行的命令，不是必须

useradd nginx

%prep

%setup -q

%build

./configure //配置，如果要添加选项或者模块可以继续写

make %{?\_smp\_mflags} //编译

%install

make install DESTDIR=%{buildroot} //安装

%files

%doc

/usr/local/nginx/\* //将改路径下文件打包成rpm

%changelog

rpmbuild -ba rpmbuild/SPECS/nginx.spec //根据上述文件制作rpm包

ls rpmbuild/RPMS/x86\_64/nginx-1.17.6-1.x86\_64.rpm //查看最终结果，

已经产生nginx的rpm包则成功

-----------------------------------------------------------------------------------

**三，VPN 虚拟专用网**

**1，使用gre技术搭建vpn，适用于linux环境**

[root@web1 ~]# modprobe ip\_gre //在内核中开启gre功能

[root@web1 ~]# lsmod | grep gre //检查，可以看到gre字样信息即可

ip tunnel add tun0 mode gre remote 192.168.2.200 local 192.168.2.100 //创建

vpn隧道，名字叫tun0，mode是使用gre技术，与2.200连接，自己ip是2.100

ip tunnel del tun0 //如果错误，可以删除重配

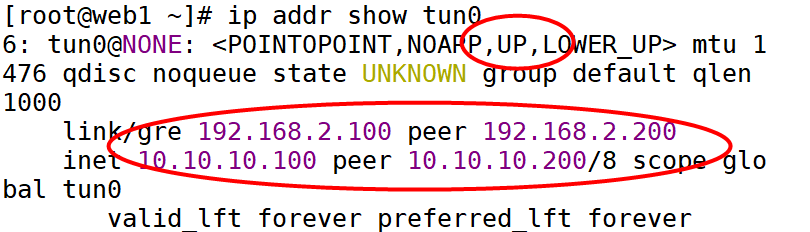
ip addr add 10.10.10.100/8 peer 10.10.10.200/8 dev tun0 //在tun0隧道(dev)中使

用私有ip地址，本机是10.100 ，对面(peer)是10.200

ip addr del 10.10.10.100/8 peer 10.10.10.200/8 dev tun0 //如果错误可以删除

ip link set tun0 up //激活tun0

ip addr show tun0 //查看



**2，使用pptpd搭建vpn，支持windows环境**

[root@web1 ~]# cd ~/lnmp\_soft/vpn/

yum -y install ppp //安装vpn依赖软件包

rpm -ivh pptpd-1.4.0-2.el7.x86\_64.rpm //安装vpn工具

vim /etc/pptpd.conf //修改102、103行

localip 192.168.2.100 //本机ip

remoteip 10.10.10.10-18 //给windows客户机分配的ip(vpn隧道内

使用的地址，可以自定义)，范围是10.10.10.10~10.10.10.18

vim /etc/ppp/options.pptpd //修改配置，定义dns，66行将#去掉

vim /etc/ppp/chap-secrets //定义windows客户机的用户名和密码

tom \* 123456 \* //另起一行创建用户tom，配置密码123456

systemctl restart pptpd //开服务

netstat -ntulp | grep pptpd //检查

然后使用windows系统设置vpn



最下面 用户名写tom 密码123456

**3，使用xl2tp搭建vpn，支持windows环境，更安全**

systemctl stop pptpd

首先第一个环节，配置加密：

[root@web1 vpn]# yum -y install libreswan //安装加密工具

[root@web1 vpn]# cp myipsec.conf /etc/ipsec.d/ //复制配置文件到

ipsec.d目录

[root@web1 vpn]# vim /etc/ipsec.d/myipsec.conf //修改配置第16行

left=192.168.2.100 //设置为本机ip

vim /etc/ipsec.secrets //修改配置，添加加密信息

192.168.2.100 %any: PSK "randpass" //另起一行，添加2.100是本机

ip，%any:是允许任何客户机连接本服务器，PSK是预共享密钥，randpass

是密码，等windows客户连接vpn服务器时需要该密码

rpm -ivh xl2tpd-1.3.8-2.el7.x86\_64.rpm

vim /etc/xl2tpd/xl2tpd.conf //打开配置文件，32、33行

ip range = 10.10.10.10-10.10.10.18 //给客户分配的ip

local ip = 192.168.2.100 //本机ip

vim /etc/ppp/options.xl2tpd //修改配置文件，将第10、16行

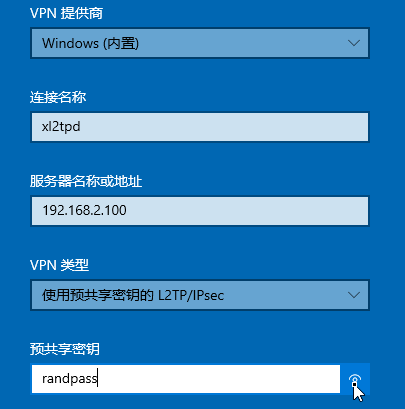
注释掉，删除掉21行的#以及空格，就可以启用加密

systemctl start ipsec //开启加密服务

netstat -ntulp |grep :500 //之后检查加密服务

systemctl start xl2tpd //开启xl2tpd服务

netstat -ntulp |grep xl2tpd



--------------------------------------------------------------------------------------------------------

**四，编写Unit文件，使systemctl命令控制nginx**

cd /usr/lib/systemd/system

cp httpd.service nginx.service //拷贝模板

vim nginx.service //修改

[Unit]

Description=The nginx HTTP Server //描述

After=network.target remote-fs.target nss-lookup.target //在网络服务、网络文件服务、域名服务的程序启动之后再启动nginx

[Service]

Type=forking //nginx是多进程类型程序，要设置为forking

ExecStart=/usr/local/nginx/sbin/nginx //当执行了systemctl start nginx之后执行的命令

ExecReload=/usr/local/nginx/sbin/nginx -s reload //当执行了systemctl reload nginx之后执行的命令

ExecStop=/bin/kill -s QUIT ${MAINPID} //当执行了systemctl stop nginx之后执行的命令，这里是用kill命令发送退出信号给nginx的进程号，相当于停止nginx服务，-s QUIT是发送退出信号，${MAINPID}是变量，里面存了nginx的进程号

[Install]

WantedBy=multi-user.target //支持开机自启

然后重启服务之后可以用systemctl等命令控制nginx

提示：必须要提前先安装好nginx服务，并且没有其他服务占用80端口！