山东大学 计算机科学与技术 学院

计算机网络 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201400130051 | 姓名：曹德福 | | 班级： 2014级3班 |
| 学号：201400130024 | 姓名：沈立凡 | | 班级：2014级3班 |
| 实验题目：基于Linux/iptables的网络访问服务器 | | | |
| 实验学时：2 | | 日期： 2016年11月10日 | |
| 实验目的：  学习Linux的NAT、iptables等网络功能；加强对网络访问控制和NAS的理解；锻炼编写bash脚本的能力。 | | | |
| 硬件环境：  系统制造商: Dell Inc.  系统型号: Inspiron 5547  系统类型: x64-based PC  处理器: 安装了 1 个处理器。  [01]: Intel64 Family 6 Model 69 Stepping 1 GenuineIntel ~1300 Mhz  BIOS 版本: Dell Inc. A04, 2014/5/9  Windows 目录: C:\WINDOWS  系统目录: C:\WINDOWS\system32  启动设备: \Device\HarddiskVolume1  物理内存总量: 3,984 MB | | | |
| 软件环境：  虚拟机：Vmware12  远程连接工具：putty  Windows系统：xp  Linux系统：debian 8x 64位，ubuntu14.0.4 | | | |
| 实验步骤与内容：   1. 实验内容   NAS实验。使用一个双网卡Linux为路由器(NAT网关)，对一个内部局域网的网络访问进行控制，内网用户需要使用帐号/口令登录路由器(网关)才能访问外网。  远程网桥实验。使用OpenVPN，经过互联网桥接两个LAN。   1. 实验步骤 2. 网络结构和网卡配置   外网网卡接口eth0，能访问互联网。  内网网卡接口eth1，对内网提供DHCP和NAT服务。提供设置如下：  cat /etc/network/interfaces  allow-hotplug eth0  iface eth0 inet static  address 211.87.235.73  netmask 255.255.255.0  gateway 211.87.235.1  dns-nameservers 202.194.15.12  #  allow-hotplug eth1  iface eth1 inet static  address 10.1.1.1  netmask 255.255.255.0  内网上用一个XP做测试机。     1. 安装和配置DHCPD   aptitude install udhcpd  vi /etc/udhcpd.conf  start 10.1.1.2  end 10.1.1.254  interface eth1  option subnet 255.255.255.0  option router 10.1.1.1  option dns 202.194.15.12  /etc/init.d/dhcdbd restart  # --or--  /etc/init.d/udhcpd restart     1. 开启NAT功能   开启IPv4路由和NAT。  echo 1 > /proc/sys/net/ipv4/ip\_forward  iptables -t nat -A POSTROUTING -s 10.1.0.0/24 -o eth0 -j MASQUERADE  至此，从内网机XP上可以自动获得IP地址，并能访问互联网。  删除NAT的命令是把-A改成-D。  iptables -t nat -D POSTROUTING -s 10.1.0.0/16 -o eth0 -j MASQUERADE           1. NAS访问控制 2. 阻止访问   在用户登录成功前，默认阻止一切访问，但是为了方便，允许登录前ping网关和dns以方便用户了解网络状况并能解析域名。  iptables -t filter -A FORWARD -s 10.1.0.0/16 -d 202.194.15.12/32 -j ACCEPT  iptables -t filter -A FORWARD -s 10.1.0.0/16 -o eth0 -j REJECT  为方便在用户浏览器中展现登录页面，其他任何TCP连接（包含HTTP）都被强制转向网关的对内地址。  iptables -t nat -A PREROUTING -s 10.1.0.0/16 -p tcp -j DNAT --to 10.1.1.1  针对每个成功登录的用户的IP 10.x.y.z执行这两条规则，则该用户的IP放行。  iptables -t nat -I PREROUTING -s 10.x.y.z/32 -j ACCEPT  iptables -t filter -I FORWARD -s 10.x.y.z/32 -o eth0 -j ACCEPT    要注意的是红框标注的位置，是提醒错误的，所以在最后将其加入到了脚本文件的注释当中。  综合起来，这些IPtables命令可以写成自动开机及运行的脚本（）：  cat /etc/rc.local  …  echo 1 > /proc/sys/net/ipv4/ip\_forward  iptables -t filter -A FORWARD -s 10.1.0.0/16 -d 202.194.15.12/32 -j ACCEPT  iptables -t filter -A FORWARD -s 10.1.0.0/16 -o eth0 -j REJECT  iptables -t nat -A PREROUTING -s 10.1.0.0/16 -p tcp -j DNAT --to 10.1.1.1  iptables -t nat -A POSTROUTING -s 10.1.0.0/16 -o eth0 -j MASQUERADE  #iptables -t nat -I PREROUTING -s 10.x.y.z/32 -j ACCEPT  #iptables -t filter -I FORWARD -s 10.x.y.z/32 -o eth0 -j ACCEPT    取消全部iptable规则的命令：  iptables -t nat -F  iptables -t filter -F  iptables -t mangle –F  做完上述步骤之后，重新启动debian系统。   1. 准备登陆页面   aptitude install apache2    在10.1.1.1（也可以是10.1.x.y）这个IP地址上有个由apache2提供的登录网页index.html(/var/www)。｛这些网页和脚本在后面附件里有。｝  vi /var/www/index.html  <html><head>  <meta http-equiv="refresh" content="0; url=http://10.1.1.1/login.htm">  </head>  <body><h1>It works!</h1></body>  </html>  login页面像这样：    另外，为了避免漏网之页面，在apache的配置文件/etc/apache2/apache2.conf中添加一项：  vi /etc/apache2/apache2.conf  ErrorDocument 404 <http://10.1.1.1/redir.htm>    Redir页面像这样：  cat redir.htm  <html><head>  <meta http-equiv="refresh" content="0; url=http://10.1.1.1/login.htm">  </head>  <body>wait 0 second...<br>稍候。。。</body>  </html>  如果测试有异常，大概需要注释掉/etc/apache2/sites-available/default 中的含有RedirectMatch的那行。   1. 登录处理/cgi-bin/checkin.cgi(/usr/lib/cgi-bin)   chmod +s `which iptables`  chmod +x /usr/lib/cgi-bin/checkin.cgi  准备用户及其口令：  cat /etc/nasuser.list  cngi passwd  user1 passwd  user2 passwd  user3 passwd    至此，可以从内网机器上访问一试了。   1. 其他变化   在内网中使用和外网一致的静态地址（而不是私有地址10.x.y.z或192.168.x.y），且不必改动上游路由器的配置。这是能做到的。  添加拨号支持。外网用户可以拨号获得内网用户身份以访问内网资源。  面对网页乱码情况，有两种解决方案，一种是在XP系统中设置浏览器编码为UTF-8，另一种是直接在apache网页部分设置编码格式为UTF-8       1. 实验运行结果如下：       至此，已经可以正常上网了：     1. 支持IPV6   （1）内网上IPV6地址前缀  IPv6地址的配置是自动的，前缀64位，拼上从MAC地址衍生来的64位后缀，即为完整的128位IPv6地址。IPv6的路由器可以使用radvd分发地址配置信息，主要是广播地址前缀。  安装radvd组件：  aptitude install radvd  比如在eth1网段上广播路由前缀2001:da8:7001:18::/64，配置文件/etc/radvd.conf内容如下。  vi /etc/radvd.conf  interface eth1 {  AdvSendAdvert on;  MinRtrAdvInterval 3;  MaxRtrAdvInterval 10;  prefix 2001:da8:7001:18::/64 {  AdvOnLink on;  AdvAutonomous on;  AdvRouterAddr on;  };  };  使用radvdump命令可以看到radvd的路由广播信息。  为简便，设置eth1的地址为2001:da8:7001:18::10.1.1.1。可以写到rc.local中去。  ifconfig eth1 add 2001:da8:7001:18::10.1.1.1/128  eth1所在网络上的其他机器可以自动获得3e79:1::/64前缀的IPv6地址，其后64位由网卡MAC地址自动衍生。比如某机器A获得的地址是3e79:1::20c:29ff:fedc:a021，机器B获得的地址是3e79:1::20c:29ff:feb4:290c。则A、B和Rose机器相互之间可以使用3e79:1::/64前缀的地址相互ping通。  查看本机IPv6地址可以使用ipconfig.exe或ifconfig。  （2） 内网IPV6路由问题  开启IPv6路由转发。写到rc.local中。  echo 1 > /proc/sys/net/ipv6/conf/all/forwarding  仅仅这样是不够的。此时IPv6分组能够发送到外网，但是返回分组回不到内网机上来。这个现象可以使用tcpdump观测验证。原因是NAS的上游路由器发出的ND请求(相当于IPv4中的ARP)被NAS阻隔，内网机器根本就收不到。  IPv6中有没有类似IPv4中的Proxy ARP的功能？内核2.6.19(2006/11)中开始有这个proxy\_ndp功能。(http://www.ipsidixit.net/2010/03/24/239/)  echo 1 > /proc/sys/net/ipv6/conf/all/proxy\_ndp  ip -6 neigh add proxy x::y/128 dev eth0  如果使用此前的内核版本，则需要自己写一个守护程序，当外网上(通常是上游的路由器)有询问内网某IPv6的MAC地址时，称其MAC是自己的eth0的MAC，这样可以截获发往内网的分组，由于forwarding已设置，自然会转发。  （3） iptable6规则  阻断IPv6分组。rc.local  ip6tables -A FORWARD -s ::/0 -j DROP  登录成功后执行  ip6tables -I FORWARD -s $ip/128 -j ACCEPT  注销时执行  ip6tables -I FORWARD -s $ip/128 -j ACCEPT  （4）IPV6下的重定向和登陆页面  引导用户使用http://[2001:da8:7001:18::10.1.1.1]，而不是<http://10.1.1.1/>去访问登录网页，即可激活访问IPv6外网。 | | | |
| 结论分析与体会：  这次实验主要练习了基于Linux/iptables的网络访问服务器，让我们懂得了类似山大登陆界面跳转等的实现原理。学习了Linux的NAT、iptables等网络功能，并加强了我对网络访问控制和NAS的理解，锻炼编写bash脚本的能力，而且使我了解了内网和外网之间的运行机制，对一些访问控制也有了进一步的理解。  虽然实现的功能还相对较简单，但是通过不断地调试和对每个文件的理解，让我们能初步独立的实现网络控制和NAS等功能，极大的锻炼了我们的自学能力以及处理错误异常的能力。这次试验同时也提醒我们设计到底层实现的时候，一定要备份相关重要文件，并且在懂得了相关原理之后再进行操作，盲目操作很容易带来系统崩溃，同时这也体现了虚拟机存在的一方面价值。 | | | |