路由网关设计报告

李俊强 201821010312

（信息与通信工程学院）

摘 要：通过在 VMware Workstation（桌面虚拟计算机软件）中安装了多台 Ubuntu（一款开源的 Linux 操作系统发行版本）虚拟机，将 Linux 操作系统配置成了一个接入网关，具备 ACL、MAC 地址绑定、“非法”IP 过滤和 PORT 过滤功能。经过测试，本测试环境成功完成了上述的所有功能。

关键词：VMware Workstation；Ubuntu；接入网关；绑定；过滤

1 设计与配置要求

将 Linux 系统配置成一个接入网关，具备 ACL、MAC 地址绑定、“非法”IP 过滤和 PORT 过滤功能。搭建一个子网接入的实验测试网络。

2 设计思路与方案

整个设计与配置过程分为三部分。第一个部分是网络环境搭建，这个是进行后续配置的基础。第二个部分是设计接入网关，这部分的总体思路比较简单。首先同时开启多台 Linux 主机，把其中的一台作为接入网关的服务器，剩余的其他主机都连接到该接入网关上，获取到各自的 IP 地址、默认网关等信息。在完成第二部分的前提下，便可以开始第三部分的设计，也就是配置网络使其具备 ACL、MAC 地址绑定、“非法”IP 过滤和 port 过滤功能。这个部分用到了 Linux 系统自带的命令：iptables。这个命令是 Linux 内核集成的，能够完成 IP 信息包过滤功能，后面我们会详细介绍这个命令。

为了更好地对配置结果进行测试，我在接入网关的服务器上搭建了两个网卡。一个网卡实现了接入网关的功能，负责对接入主机进行 IP 地址分配；另一个网卡实现了 NAT 功能，负责与因特网进行连接。整个实验测试网络的示意图如图 1 所示。因此，我们可以通过判断某个接入主机是否具有上网功能来确定接入网关和过滤功能是否配置成功。

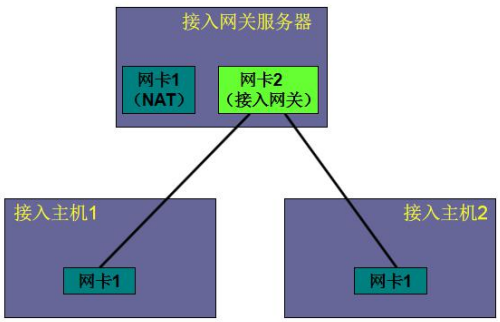

图 1 测试网络示意图

3 软件选型

安装 Linux 操作系统有两种方法：一种方法是直接在电脑上安装，如果之前电脑已经有一个操作系统（Windows 或者 Linux），那么新安装的系统会与原有的系统独立，两者互不干扰，但每次只能同时开一个操作系统；另一种方法是在虚拟机上安装，这种方法可以同时运行多个系统，调试起来非常方便。

因为我们需要搭建多台 Linux 主机的测试网络环境，如果采用第一种方法，需要用到多台电脑，而且每台电脑上都装一个 Linux 操作系统，实现成本高而且配置也比较繁琐。因此，我采用了第二种方法来完成此次配置。这种方法也有一个缺点，多个操作系统同时运行时非常耗系统内存。因此，为了能让我的电脑同时带动多个 Linux 系统运行，我安装了 Ubuntu16.04 服务器版本。服务器版相比于桌面版，少了图形化用户界面，可以大大降低系统内存的使用。而且配置过程中，其实用不到图形化用户界面，因此选用该版本是合理的。整个测试网络的软件选型结果如表 1 所示。

表 1 软件选型结果

虚拟机软件	Linux 版本
VMware Workstation 12	Ubuntu 16.04 服务器版

4 配置方案

整个配置方案分为 3 个部分：第一个部分是网络环境搭建，第二个部分是接入网关的配置，第三个部分是绑定与过滤功能配置。下面，我将分别对这三个部分进行详细地描述。

4.1 网络环境搭建

因为本次配置是在 VMware Workstation 虚拟机上进行的，所以我们首先得了解该虚拟机的网络配置模式，这样才能更好地了解整个网络的内部逻辑结构。同时，在分析了网络配置模式之后，我们对本次搭建的网络环境进行详细地描述。

4.1.1 VMware Workstation 网络配置模式

VMware Workstation 虚拟机有 4 种网络配置模式，分别是：桥接模式、NAT 模式、仅主机模式和自定义模式。

桥接模式（Bridge）：在 VMware Workstation 中，桥接模式默认使用 VMnet0 网卡。在这种模式下，真正的本地物理网卡会与虚拟机 VMnet0 网卡通过链路层协议进行互连，并基于链路地址选择要传递的数据。实际这种方式就是直接连接到物理网络上的一种连接方式，适合同网段可以获取多个 IP 地址的情况，如图 2 所示。

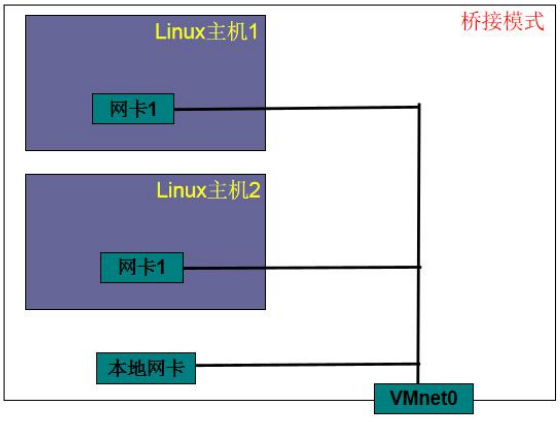

图 2 桥接模式示意图

NAT 模式（Network Address Translation）：在 VMware Workstation 中，NAT 模式默认使用 VMnet8 网卡。NAT，即网络地址转换，能够将私有地址转化为合法 IP 地址，广泛应用于各种类型 Internet 接入方式和各种类型的网络中。NAT 连接方式不仅解决了 IP 地址不足的问题，而且还能够避免来自网络外部的恶意攻击，隐藏并保护网络内部的计算机系统。其连接方式如图 3 所示。

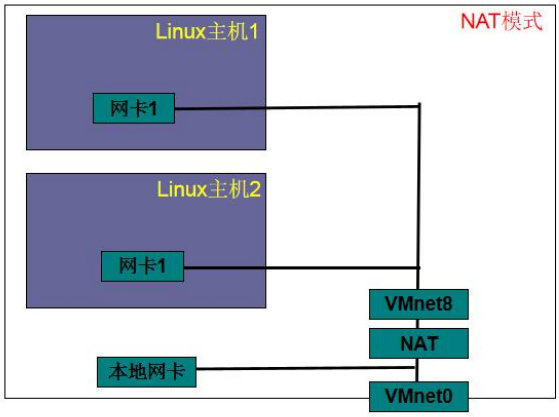

图 3 NAT 模式示意图

仅主机模式（Host-Only）：在 VMware Workstation 中，仅主机模式默认使用 VMnet1 网卡。这种模式是虚拟机与本地物理机之间进行的网络互访，也就是说，通过虚拟机只能够访问到本地物理机，而不能够让虚拟机访问 Internet。其连接方式如图 4 所示。

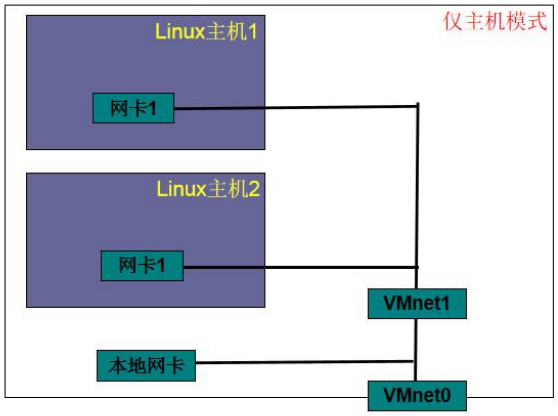

图 4 仅主机模式示意图

自定义模式：通过自定义方式可以指定 10 个虚拟交换机 VMnet0~VMnet9 中任意一个。除了上述讲解的虚拟机三个默认网卡：VMnet0、VMnet1、VMnet8，我们还可以根据需要添加 VMnet2~VMnet7 和 VMnet9 等 7 个虚拟交换机。

4.1.2 网络环境配置方案

为了保证此次配置更具有真实性，我安装了 4 个 Ubuntu16.04 服务器版的操作系统。其中 1 个系统作为接入网关服务器，剩下的 3 个作为接入主机。同时，为了更方便地对配置的正确性进行验证，我在网关服务器中添加了两个网卡，一个网卡实现了接入网关的功能，另一个网卡实现了 NAT 功能。不仅如此，为了让接入主机不受到其他网卡的干扰，我把接入主机的网卡和接入网关的网卡都同时设定为自定义模式（VMnet7），这个模式下关闭了桥接、NAT 和 DHCP 功能。VMnet7 的设置如图 5 所示。

综合上述分析，我们可以得出整个网络环境的模式配置方案，如表 2 所示，同时也能够得到整个网络环境的内部逻辑结构，如图 6 所示。

表 2 网络环境模式配置方案

主机名	数量（台）	网卡数目（个）	网卡模式（网卡名）
接入网关服务器	1	2	NAT 模式（VMnet8）、 自定义模式（VMnet7）
接入主机	3	每个主机各 1 个	自定义模式（VMnet7）

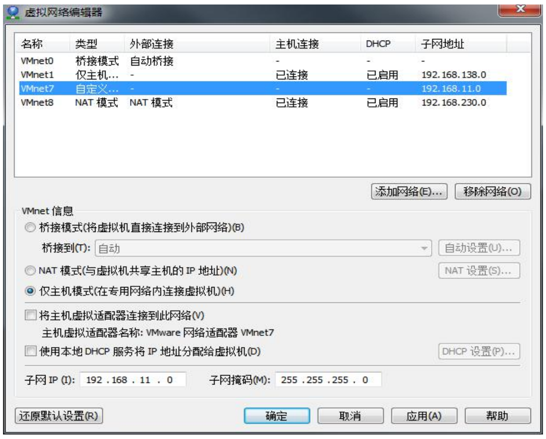

图 5 自定义模式（VMnet7）的设置

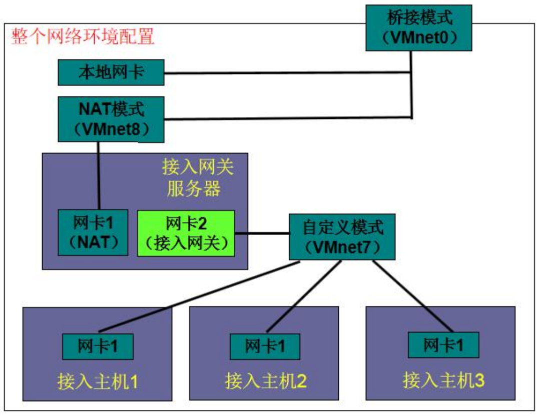

图 6 整个网络环境的内部逻辑结构

4.2 接入网关的配置

接入网关（AG，Access Gateway），顾名思义，也就是提供网络接入功能。一定程度来说，接入网关其实相当于一个 DHCP 服务器，它的内部配置了一个 DHCP 地址池，连接到该服务器的主机将会分配到地址池当中的 IP 地址、默认网关等资源。因此配置的第一步就要在服务器当中配置 DHCP 服务器。值得注意的是，为了不让虚拟机 VMware Workstation 的默认网卡干扰 DHCP 的配置，必须将自定义模式的网卡 VMnet7 的 DHCP 功能关闭，否则接入主机将会默认从 VMnet7 中获取 IP 地址，而不是从接入网关服务器当中获取。

4.2.1 配置静态 IP 地址

在接入网关服务器配置 DHCP 服务之前，首先要保证自身的网卡上有一个静态 IP 地址。配置静态 IP 地址的过程如下所示。

1. 首先使用 vim 编辑器打开/etc/network/interfaces，命令如下:

sudo vim /etc/network/interfaces

2. 然后往里面输入如下代码：

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet static
address 192.168.11.1
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255

auto ens38
iface ens38 inet dhcp
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/network/interfaces" 19L, 431C                          1,1           All
```

注意：网关服务器有两个网卡，一个是ens33，作为接入网关；另一个是ens38，作为NAT网关。因为ens33连接的是VMnet7（见图6），而VMnet7的子网地址为192.168.11.0（见图5），所以ens33网卡设置的静态地址必须在该网段内，本次配置中我设置为192.168.11.1。

3. 配置后，重启网卡使其生效，命令如下：

sudo /etc/init.d/networking restart

4. 查看静态IP地址是否配置成功，命令为：ifconfig。

5. 配置成功后，可以看到如下情况：

```
li@ubuntu:~$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:f1:ab:0b
          inet addr:192.168.11.1  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef1:ab0b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:486 (486.0 B)  TX bytes:1226 (1.2 KB)

ens38     Link encap:Ethernet  HWaddr 00:0c:29:f1:ab:15
          inet addr:192.168.230.143  Bcast:192.168.230.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef1:ab15/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4649 (4.6 KB)  TX bytes:5476 (5.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:49 (49.0 B)  TX bytes:49 (49.0 B)

li@ubuntu:~$
```

4.2.2 配置 DHCP 服务器

配置完静态IP地址后，我们便可以开始配置DHCP服务器了，配置过程如下所示。

1. 安装DHCP服务，命令如下：

sudo apt-get install isc-dhcp-server

2. 配置DHCP服务。

I. 首先用vim编辑器打开/etc/default/isc-dhcp-server文件，命令如下：

sudo vim /etc/default/isc-dhcp-server

II. 然后，对里面的内容修改如下：

```
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="ens33"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/default/isc-dhcp-server" 21L, 658C                      1,1           All
```

III. 然后用vim编辑器打开 /etc/dhcp/dhcpd.conf 文件，命令如下：

sudo vim /etc/dhcp/dhcpd.conf

IV. 最后，往该文件添加如下内容：

```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

#subnet 10.254.239.0 netmask 255.255.255.224 {
#  range 10.254.239.10 10.254.239.20;
#  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}
subnet 192.168.11.0 netmask 255.255.255.0 {
   range 192.168.11.20 192.168.11.40;
   option routers 192.168.11.1;
   option subnet-mask 255.255.255.0;
   option broadcast-address 192.168.11.255;
   option domain-name-servers 192.168.11.1;
   option ntp-servers 192.168.11.1;
   option netbios-name-servers 192.168.11.1;
   option netbios-node-type 8;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
                                                          57,1          24%
```

4.2.3 配置 NAT 服务

前面我们说到，接入网关服务器有两个网卡，一个是提供接入网关服务（ens33），一个是提供NAT服务（ens38）。因此，我们可以把ens38网卡的IP地址当做是一个公网地址，要想接入主机能够上网，就必须为ens38网卡配置NAT，这样接入主机就能够通过接入网关服务器的ens38与虚拟机的NAT网卡通信，进而访问Internet。

配置NAT服务的命令只要一行，如下所示：

sudo iptables -t nat -A POSTROUTING -s 192.168.11.0/24 -o ens38 -j MASQUERADE

4.2.4 配置接入网关服务器的转发功能

因为接入网关服务器有两个网卡，要让连接到服务器ens33网卡的接入主机发送的数据包能够通过服务器的ens38网卡传递到外界，必须要打开服务器的路由转发功能。配置过程如下所示。

1. 使用vim编辑器打开/etc/sysctl.conf文件，命令如下：

sudo vim /etc/sysctl.conf

2. 将“net.ipv4.ip_forward=1”所在的那一行“#”去掉，如下所示：

```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
"/etc/sysctl.conf" 60L, 2083C                                 30,1          Top
```

4.2.5 配置接入主机的 DNS 地址

为了能够让接入主机上网还需要最后一步：配置接入主机的DNS地址。以其中一个接入主机为例（其他两个接入主机的配置过程完全一致），配置过程如下所示：

1. 使用vim编辑器打开/etc/network/interfaces，命令如下：

sudo vim /etc/network/interfaces

2. 往该文件中添加DNS地址，如下所示：

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet dhcp
dns-nameserver 8.8.8.8
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/network/interfaces" 13L, 329C                           1,1           All
```

4.3 绑定与过滤功能配置

在第 2 部分设计思路与方案中，我们讲到 Linux 系统自身提供了 IPTABLES 工具来设置、维护和检查 Linux 内核的 IP 包过滤规则。而且，在 4.2.3 节配置 NAT 服务时，我们就用到了 iptables 命令。因此在配置绑定与过滤功能之前，我们首先得了解 IPTABLES 这个工具。

4.3.1 IPTABLES 工具

IPTABLES 工具通过分析数据包的报头数据，并与预先定义的规则来决定该数据包是否可以进入主机或者被丢弃。其中，“规则”是指管理员出于某些考虑预先设定的访问条件。也就是说，IPTABLES 工具会将每个数据包的报文头部依次地与预先定义的规则进行比对，然后执行相应的动作，以此达到过滤功能。

IPTABLES 工具设定了不同的规则表(table)和链(chain)来管理不同的规则。IPTABLES 中的表有 4 个，分别是 Filter 表、Nat 表、Mangle 表和 Raw 表。其中，Filter 表主要用于包过滤，Nat 表用于网络地址转换，Mangle 表用于重新封装报文，Raw 表用于连接追踪。

每个表都有各种不同的链。本次实验当中，我们仅用到了其中的 Filter 表和 Nat 表，因此我们只介绍这两种表的情况，如图 7 所示。

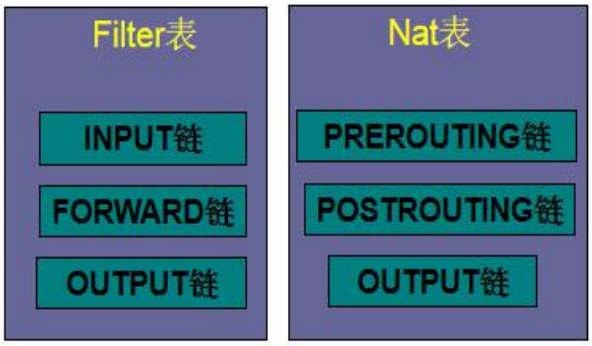

图 7 Filter 表和 Nat 表包含的链规则

我们知道，网络当中有非常多的过滤规则，当把这些规则串到一起的时候，就形成了“链”。每当有一个数据报文经过了设定规则的网络（即防火墙）上时，就需要将这条“链”上的所有规则过滤一遍，从而执行相应的动作。以其中的 INPUT 链和 PREROUTING 链为例，它们对应的内部结构如图 8 所示。对于不同的链，其对应的功能也不一样。我们仅分析与本次实验有关的链，如表 3 所示。

图 8 INPUT 链和 PREROUTING 链的内部结构

表 3 部分链对应的功能

表名	功能
INPUT	与想进入 Linux 主机的数据包相关
OUTPUT	与 Linux 主机送出的数据包相关
FORWARD	转发数据包到后端的主机
PREROUTING	在路由判断前进行的规则处理
POSTROUTING	在路由判断后进行的规则处理

在了解了上述基础之后，我们就能够使用 IPTABLES 的相关命令了。下面我将对本次实验当中用到的具体命令进行分析。

1. 删除已有的规则：iptables -F

2. 设置某个链 XXX 的默认策略为 DROP（丢弃）：iptables -P XXX DROP

具体地，如果该链的名称是 INPUT，则对应的命令为：iptables -P INPUT DROP

显然，该命令的作用是丢弃所有进入到 Linux 主机的数据包。

类似地，我们也可以设置某个链的默认策略为 ACCEPT（接收）：iptables -P XXX ACCEPT

3. 过滤“非法”源 IP XXX 和目的 IP YYY 的数据包：

iptables -A INPUT -s XXX -d YYY -j DROP

该命令的作用就是让所有进入到 Linux 主机的、源 IP 为 XXX、目的 IP 为 YYY 的数据包都进行丢弃，也就是过滤了“非法”IP。

更深入地，我们可以过滤从某个网卡（如 ens33）进入的、源 IP 为 XXX 的 TCP 数据包：

iptables -A INPUT -i ens33 -p tcp -s x.x.x.x -j DROP

4. 过滤“非法”PORT 的数据包：

iptables -A INPUT -i eth33 -p tcp --sport 1:1023 --dport 1:1023 -j DROP

该命令的作用是让所有从 ens33 进入到 Linux 主机的、源端口为 1~1023、目的端口为 1~1023 的数据包进行丢弃，也就实现了过滤“非法”PORT。

5. 配置 NAT 功能：

iptables -t nat -A POSTROUTING -s 192.168.11.0/24 -o ens38 -j MASQUERADE

该命令的作用是让所有源 IP 为 192.168.11.0/24 的网段的数据包在经过 ens38 这张网卡的时候，在路由判断后设置 nat 表让该网段的数据包进行伪装（MASQUERADE），从而实现 NAT 的功能。

6. 查看 Filter 表的 iptables 规则，以数字形式显示：

iptables -L -n

4.3.2 ACL、MAC 地址绑定配置

ACL（Access Control List，访问控制列表），能够用来控制某些数据包进出端口。ACL 通过帮助用户定义一组权限规则，说明什么样的数据包才能够对其访问，其实就是用上节讲述的 IPTABLES 命令对其进行配置。

MAC 地址绑定，我的理解是它属于 ACL 当中的一个具体的实现策略。通过 IP 与 MAC 地址绑定，使得数据包只有同时匹配到 ACL 规则设定的 IP 和 MAC 地址才能转发，能有效地减少 ARP 攻击，从而提高整个网络的安全性。

在接入网关服务器上配置 IP 与 MAC 地址绑定的 ACL 策略过程如下：

1. 打开接入主机 1，查看其 IP 和 MAC 地址，命令为：ifconfig。

我们可以清楚地看到接入主机 1 的 IP 地址为 192.168.11.20，MAC 地址为 00:0c:29:6b:cb:c5。

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:147 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22028 (22.0 KB)  TX bytes:14716 (14.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```

2. 设置接入网关服务器的 FORWARD 链默认策略为 DROP，命令如下：

sudo iptables -P FORWARD DROP

3. 设置仅允许接入主机 1 的数据包通过，也就是仅匹配接入主机 1 的 IP 与 MAC 地址，实现 MAC 地址绑定功能。在接入网关服务器输入如下命令：

sudo iptables -A FORWARD -s 192.168.11.20 -m mac --mac-source 00:0c:29:6b:cb:c5 -j ACCEPT

因为 PING 过程中，接入主机除了需要发送数据包外，也要接收对应的数据包，因此我们也要设置目的 IP 为接入主机 1 的数据包都进行接收。在接入网关服务器输入如下命令：

sudo iptables -A FORWARD -d 192.168.11.20 -j ACCEPT

4.3.3“非法”IP 过滤配置

“非法”IP 过滤是指如果某个数据包的源 IP 为规则中设定的“非法”地址，则将该数据包进行丢弃。（假设接入主机 1 的 IP 为“非法”IP）配置过程如下所示：

1. 在接入网关服务器中删除已有规则，命令为：sudo iptables -F

2. 设置接入网关服务器的 FORWARD 链默认策略为 ACCEPT，命令如下：

sudo iptables -P FORWARD ACCEPT

3. 配置“非法”IP 过滤功能，在接入网关服务器中输入如下命令：

sudo iptables -A FORWARD -s 192.168.11.20 -j DROP

4.3.4 “非法”PORT 过滤配置

“非法”PORT 过滤是指如果某个数据包的 PORT 为规则中设定的“非法”PORT，则将该数据包进行丢弃。（假设“非法”PORT 号为 2500）配置过程如下所示：

1. 在接入网关服务器中删除已有规则，命令为：sudo iptables -F

2. 配置“非法”PORT 过滤功能，在接入网关服务器中输入如下命令：

sudo iptables -A INPUT -p tcp --dport 2500 -j DROP

5 测试方法与结果分析

5.1 接入网关测试方法及结果分析

按照 4.2 小节讲述的接入网关配置方法，我们现在已经实现了接入网关的功能，同时也能让每个接入主机连接到 Internet。测试方法很简单，我们同时打开接入网关服务器以及 3 台接入主机，然后执行如下步骤（以其中第一个接入主机为例，其余两个操作完全一样）：

1. 接入主机上查看自己是否获取到了接入网关服务器的 IP 地址，输入命令：ifconfig。

2. 然后在接入主机上查看自己的默认路由是否是接入网关的 IP 地址，输入如下命令：

route -n

这两次操作的结果如下所示：我们可以看到其中一个主机的IP地址为192.168.11.20(正是配置/etc/dhcp/dhcpd.conf中设置的第一个IP地址)，而且默认路由也是接入网关的IP地址192.168.11.1。说明DHCP服务配置成功。

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3949 (3.9 KB)  TX bytes:3710 (3.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.11.1    0.0.0.0         UG    0      0        0 ens33
192.168.11.0    0.0.0.0         255.255.255.0   U     0      0        0 ens33
li@ubuntu:~$ _
```

3. 最后测试接入主机是否能够上网，输入如下命令：

ping www.baidu.com

结果如下所示：（能够有返回的时延值，即说明配置成功）

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2287 (2.2 KB)  TX bytes:2798 (2.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ ping www.baidu.com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
64 bytes from 104.193.88.77: icmp_seq=4 ttl=127 time=296 ms
64 bytes from 104.193.88.77: icmp_seq=6 ttl=127 time=297 ms
64 bytes from 104.193.88.77: icmp_seq=9 ttl=127 time=296 ms
64 bytes from 104.193.88.77: icmp_seq=10 ttl=127 time=296 ms
64 bytes from 104.193.88.77: icmp_seq=12 ttl=127 time=296 ms
64 bytes from 104.193.88.77: icmp_seq=13 ttl=127 time=295 ms
```

5.2 绑定与过滤功能测试方法及结果分析

5.2.1 ACL、MAC 地址绑定测试方法及结果分析

在完成接入网关的功能后，我们使用 4.3.2 节中讲解的 ACL、MAC 地址绑定的方法进行了配置，现在我们来测试该功能是否完成。

1. 步骤 2 设置了接入网关服务器的 FORWARD 链默认策略为 DROP，测试其是否配置成功可以使用 iptables 规则查看命令进行判断，在接入网关服务器中输入：

sudo iptables -L -n

我们可以得到如下的结果（FORWARD 链的策略已经为 DROP）：

```
li@ubuntu:~$ sudo iptables -P FORWARD DROP
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$
```

此时，选取其中的任意一台接入主机去 PING 百度，发现无法 PING 通，如下图所示：

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:306 errors:0 dropped:0 overruns:0 frame:0
          TX packets:275 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:42698 (42.6 KB)  TX bytes:33058 (33.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ ping www.baidu.com
ping: unknown host www.baidu.com
```

2. 步骤 3 设置了仅允许接入主机 1 的数据包通过。测试其是否配置成功同样可以使用 iptables 规则查看命令进行判断，在接入网关服务器中输入：

sudo iptables -L -n

我们可以看到如下结果：

```
li@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.11.20 -m mac --mac-source 00:0c:29:6b:cb:c5 -j ACCE
PT
li@ubuntu:~$ sudo iptables -A FORWARD -d 192.168.11.20 -j ACCEPT
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.11.20       0.0.0.0/0           MAC 00:0C:29:6B:CB:C5
ACCEPT     all  --  0.0.0.0/0           192.168.11.20

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$
```

同样地，我们也可以让接入主机 1 去 PING 百度，结果如下所示。说明我们已经实现了 ACL、MAC 地址绑定功能。

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:322 errors:0 dropped:0 overruns:0 frame:0
          TX packets:287 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45080 (45.0 KB)  TX bytes:34494 (34.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ ping www.baidu.com
PING www.a.shifen.com (111.13.100.91) 56(84) bytes of data.
64 bytes from 111.13.100.91: icmp_seq=1 ttl=127 time=43.2 ms
64 bytes from 111.13.100.91: icmp_seq=2 ttl=127 time=40.0 ms
64 bytes from 111.13.100.91: icmp_seq=3 ttl=127 time=46.7 ms
64 bytes from 111.13.100.91: icmp_seq=4 ttl=127 time=39.3 ms
64 bytes from 111.13.100.91: icmp_seq=5 ttl=127 time=39.3 ms
```

5.2.2 “非法”IP 过滤测试方法及结果分析

在测试完 ACL、MAC 地址绑定的功能后，我们使用 4.3.3 节中讲解的“非法”IP 过滤方法进行了配置，现在我们来测试该功能是否完成。

1. 在接入网关服务器中删除已有规则，使用 iptables 规则查看命令判断是否成功，如下所示（发现已经没有了 5.2.1 中配置的规则）：

```
li@ubuntu:~$ sudo iptables -F
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$
```

2. 设置接入网关服务器的 FORWARD 链默认策略为 ACCEPT，使用 iptables 规则查看命令判断是否成功，如下所示：

```
li@ubuntu:~$ sudo iptables -P FORWARD ACCEPT
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$ _
```

同样地，我们也可以让接入主机 1 去 PING 百度，结果如下所示，说明已有规则已经删除成功。

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:361 errors:0 dropped:0 overruns:0 frame:0
          TX packets:323 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49975 (49.9 KB)  TX bytes:38418 (38.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ ping www.baidu.com
PING www.wshifen.com (103.235.46.39) 56(84) bytes of data.
64 bytes from 103.235.46.39: icmp_seq=1 ttl=127 time=70.7 ms
64 bytes from 103.235.46.39: icmp_seq=2 ttl=127 time=70.8 ms
64 bytes from 103.235.46.39: icmp_seq=3 ttl=127 time=70.3 ms
```

3. 配置“非法”IP 过滤功能，测试其是否配置成功可以使用 iptables 规则查看命令进行判断，在接入网关服务器中输入：

sudo iptables -L -n

结果如下所示（FORWARD 链中已经有我们设定的规则）：

```
li@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.11.20 -j DROP
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.11.20        0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$
```

同样地，我们也可以让接入主机 1 去 PING 百度，结果如下所示，说明已经过滤掉接入主机 1（“非法”IP）的数据包。

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:392 errors:0 dropped:0 overruns:0 frame:0
          TX packets:350 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53683 (53.6 KB)  TX bytes:41356 (41.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ ping www.baidu.com
ping: unknown host www.baidu.com
li@ubuntu:~$ _
```

5.2.3 “非法”PORT 过滤测试方法及结果分析

在测试完“非法”IP 过滤的功能后，我们使用 4.3.4 节中讲解的“非法”PORT 过滤方法进行了配置，现在我们来测试该功能是否完成。

测试过程当中，我们需要用到一款网络性能测试工具——iperf。该工具能够指定本机以 server 模式或者 client 模式启动指定 TCP 或者 UDP 端口，而且还可以测试 TCP 或者 UDP 带宽质量。下面我们开始对“非法”PORT 过滤功能进行测试：

1. 在接入网关服务器中删除已有规则，使用 iptables 规则查看命令判断是否成功，如下所示（发现已经没有了 5.2.2 中配置的规则）：

```
li@ubuntu:~$ sudo iptables -F
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$ _
```

2. 在接入网关服务器上以 server 模式开启 1 个 TCP 服务器端口：2500，命令如下所示：

iperf -s -p 2500

输入后结果如下：

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:f1:ab:0b
          inet addr:192.168.11.1  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef1:ab0b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:592 errors:0 dropped:0 overruns:0 frame:0
          TX packets:437 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:73861 (73.8 KB)  TX bytes:145270 (145.2 KB)

ens38     Link encap:Ethernet  HWaddr 00:0c:29:f1:ab:15
          inet addr:192.168.230.143  Bcast:192.168.230.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef1:ab15/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:471 errors:0 dropped:0 overruns:0 frame:0
          TX packets:308 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:154522 (154.5 KB)  TX bytes:36400 (36.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:49 (49.0 B)  TX bytes:49 (49.0 B)

li@ubuntu:~$ iperf -s -p 2500
------------------------------------------------------------
Server listening on TCP port 2500
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

此时，在接入主机上以 client 模式向接入网关服务器的 TCP 端口发送数据，命令如下：

iperf -c 192.168.11.1 -p 2500

输入后结果如下所示（接入主机可以向接入网关服务器的 2500 端口发送数据）：

```
li@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:6b:cb:c5
          inet addr:192.168.11.20  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6b:cbc5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5558 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1124315 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:451431 (451.4 KB)  TX bytes:3202247412 (3.2 GB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

li@ubuntu:~$ iperf -c 192.168.11.1 -p 2500
------------------------------------------------------------
Client connecting to 192.168.11.1, TCP port 2500
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.11.20 port 46906 connected with 192.168.11.1 port 2500
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.47 GBytes  1.26 Gbits/sec
li@ubuntu:~$
```

3. 在接入网关服务器中配置“非法”PORT（2500 端口）过滤功能后，使用 iptables 规则查看命令判断是否成功，如下所示（INPUT 链中添加了对应的规则）：

```
li@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 2500 -j DROP
li@ubuntu:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:2500

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
li@ubuntu:~$ _
```

同样地，我们也可以让接入主机 1 再去向网关服务器的 2500 端口发送数据，结果如下所示，说明已经过滤掉非法 PORT（2500 端口）的数据包。

```
li@ubuntu:~$ iperf -c 192.168.11.1 -p 2500
connect failed: Connection timed out
li@ubuntu:~$
```

6 总结

本次实验是在 VMware Workstation 虚拟机中完成的。经过测试，我成功地将 Linux 操作系统配置成了一个接入网关，而且具备 ACL、MAC 地址绑定、“非法”IP 过滤和 PORT 过滤功能，实现了本次设计与配置的全部要求。经过这次的配置，我对 Linux 操作系统，网络协议如 NAT 协议、DNS 协议、DHCP 协议等，以及防火墙都有了更加深刻的认识。

参考文献:

[1]芮雪.虚拟机下Linux操作系统的网络配置[J].电脑与信息技术,2011,19(06):7-8+18.

[2]谢希仁. 计算机网络[M].电子工业出版社, 2008.1.

[3]鸟哥. 鸟哥的Linux私房菜——服务器假设篇[M].机械工业出版社，2012.6.