Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий Механики и Оптики

Факультет Информационных Технологий и Программирования

Лабораторная работа №7

"Конфигурирование межсетевого экрана и NAT"

Выполнил: Мыльников Александр

Группа: 3511

Санкт-Петербург

2014

Цель: Сформировать понимание принципов работы NAT и firewall, а так же начальные навыки в конфигурировании NAT и Firewall на платформе Windows (routing and remote access service) и Linux (iptables).

Необходимо:

1) Установленная на компьютере среда виртуализации ORACLE Virtual Box

2) Образы виртуальных жёстких дисков операционных систем Windows 2003 и Linux

3) Доступ к сети Интернет

4) Учетные записи пользователей с администраторскими правами

Краткие теоретические сведения:

NAT (Network Address Translation) – технология стека TCP\IP. Она позволяет модифицировать заголовки пересылаемых через NAT IP-пакетов и TCP\UDP сообщений. NAT в общем случае представляет собой компьютер или аппаратный маршрутизатор, подключенный одним интерфейсом к внешней сети, а другими к внутренней. Оба интерфейса имеют IP адреса в каждой из сетей. Типичным применением NAT является обеспечение доступа из локальной сети с приватными IP-адресами к ресурсам внешней сети с IP-адресами интернет. При передаче запроса от локального клиента к внешнему ресурсу подменяется сокет отправителя: IP адрес меняется на внешний IP адрес NAT, а порт на свободный порт на внешнем интерфейсе NAT. Когда приходит ответ от внешнего ресурса, происходит обратная замена сокета и пакет передается в локальную сеть полкучателю. Так же с помощью NAT можно публиковать локальные сокеты на реальном IP адресе и реальном порту. Например для обеспечения доступа извне к Web серверу, расположенному в локальной сети. В этом случае на NAT делается статическое отображение внешнего сокета на внутренний.

Под межсетевым экраном или брандмауэром понимают фильтр IP пакетов предназначенный для формального ограничения соединений клиентов и серверов работающих «поверх» стека TCP\IP.

В основу работы классического firewall положен контроль формальных признаков. В общем случае фильтрация осуществляется по:

IP адресам отправителя и получателя в заголовке IP пакета

номерам портов приложения-получателя и приложения-отправителя

инкапсулированным в IP протоколам транспортного (TCP, UDP) и сетевого

уровней (ICMP).

Правила фильтрации формируются в виде списка. Все проходящие пакеты проверяются

по списку последовательно, до первого срабатывания. Последующие правила к пакету не

применяются.

Для конфигурирования firewall в Linux необходимо сформировать набор правил

iptables. В iptables реализовано несколько таблиц (filter, nat и т.д.). Каждая таблица

содержит набор цепочек правил. Например цепочки INPUT для входящего трафика,

OUTPUT для исходящего, FORWARD для пересылаемого, PREROUTING для обработки

трафика перед пересылкой и т.д.. Набор цепочек для разных таблиц различен.

Управление цепочками производится с помощью консольной команды iptables.

Примеры:

iptables -t filter -A INPUT -s ws.mytrust.ru -j ACCEPT включает прием всех

пакетов с хоста ws.mytrust.ru

iptables -t filter -A OUTPUT -d mail.ifmo.ru --dport 25 -j DROP запрещает отправку

всех пакетов на хост mail.ifmo.ru на порт 25

iptables -t filter -A INPUT -j DROP запрещает прием всех сообщений2

В протоколах TCP и UDP (семейства TCP/IP) порт — идентифицируемый номером

системный ресурс, выделяемый приложению, выполняемому на некотором сетевом хосте,

для связи с приложениями, выполняемыми на других сетевых хостах (в том числе с

другими приложениями на этом же хосте).

На платформе Windows Server NAT настраивается с помощью службы Routing and Remote

Access. В Windows 2003 в консоли этой службы можно настроить и правила фильтрации

для передаваемого трафика. Для защиты конечных хостов в Windows служит отдельная

служба windows firewall. Запуск этих служб на windows 2003 одновременно невозможен.

В старших версиях Windows существует отдельный общий firewall – advfirewall, для всех случаев.

На платформе Linux и для настройки NAT и для фильтрации трафика используется iptables. Управление iptables осуществляется или из командной строки (см. выше) или через конфигурационный файл /etc/sysconfig/iptables (для CentOS и др. дистрибутивов семейства RedHat). Важно отметить, что для того чтобы Linux начал пересылать пакеты из интерфейса в интерфейс надо чтобы в параметре ядра net.ipv4.ip_forward = 1. Установить его можно с помощью утилиты sysctl, или записью в конфигурационный файл в каталоге /proc.

В серде виртуал бокс существут 2 машины. они находятся за натом. внутренним дропбокса. Шлюз получает адрес по dhcp к нему прилетает адрес внешнего днс родительского роутера.

линукс Unbuntu (gateway) + Unubru user host
Настройки первой машны.
Для того чтобы машина работала в виде шюза она должна иметь два интерфейса. WAN который смотрит в NAT Virtual Box и LAN интерфейс по заданию 10.0.0.0/8
Конфиги /etc/network/interfaces

```
auto lo
iface lo inet loopback

auto WAN
iface WAN inet dhcp  #(сетка NAT VirtualBox)
auto LAN
iface LAN inet static
  address 10.0.0.1
  netmask 255.255.255.0
```

Вторая важная задача на хосте разрешить форвардинг IP
файл /etc/sysctl.conf

```
net.ipv4.ip_forward=1
```

На “клиентском хосте ” (я выбрал Ubuntu, да простите вы меня уважемый преподаватель. Искренне, ничего личного. Мне сильно нужна практика в настройке линукс. От настройки виндовс у меня несварение)
важен /etc/network/interfaces здесь мы стически задаем ip для интерфейса eth0

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
```

```
  address 10.0.0.2
  gateway 10.0.0.1
  netmask 255.255.255.0
  dns-nameservers 192.168.1.1 #самый корневой роутер сети.
```

Итого у нас получился нат виртуалбокса. который выдает ип и инет для гейтвея, далее гейтвей как роутер общается с машиной. начинаем проброс портов
Собственно настройка Firewall на роутере

```
iptables --policy INPUT DROP
iptables --policy OUTPUT DROP
iptables --policy FORWARD DROP
iptables -F
iptables -t nat -F
iptables -t mangle -F

iptables -X
iptables -t nat -X
iptables -t mangle -X
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -p tcp --dport ssh -j ACCEPT
iptables -A PREROUTING -t nat -i WAN -p tcp --dport 2222 -j DNAT --to 10.0.0.2:22
iptables -A FORWARD -p tcp -d 192.168.1.2 --dport 22 -j ACCEPT
iptables -A OUTPUT -p udp -o LAN --dport 53 -d 192.168.1.1 -j ACCEPT #разрешить логальный DNS
iptables -A INPUT -p udp -i LAN --sport 53 -d 192.168.1.1 -j ACCEPT
iptables -A INPUT -i LAN -p icmp --icmp-type echo-request -j DROP #block ping

iptables -A  OUTPUT -i LAN -p tcp --dport 143 -d 194.85.160.50 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -o LAN -p tcp --sport 143 -d 194.85.160.50 -m state --state ESTABLISHED -j ACCEPT #imap
iptables -A OUTPUT -i LAN -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A  INPUT  -o LAN -p tcp --sport 110 -m state --state ESTABLISHED -j ACCEPT #pop3

iptables -A OUTPUT -i LAN -p tcp --dport 8080 -d 194.85.160.55 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A  INPUT  -o LAN -p tcp --sport 8080 -d 194.85.160.55  -m state --state ESTABLISHED -j ACCEPT #http proxy

iptables -A OUTPUT -i LAN -p tcp --dport 21 -d 194.85.160.60 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A  INPUT  -o LAN -p tcp --sport 21 -d 194.85.160.60  -m state --state ESTABLISHED -j ACCEPT #http proxy
iptables -A INPUT -i WAN -d 10.10.11.173 DROP
iptables -A INPUT -i LAN-d 10.10.11.173 DROP
iptables -A OUTPUT -i WAN -d 10.10.11.173 DROP
```

```
iptables -A OUTPUT -i WAN 10.10.11.173 DROP
iptables -A INPUT -i WAN -p tcp --dport 22 -d 83.0.0.0/16 -m state --state
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o WAN -p tcp --sport 22 -d 83.0.0.0/16 -m state --state ESTABLISHED
-j ACCEPT #accept ssh  83.0.0.0/16

все это должно быть сохранено /etc/iptables.rules
sudo sh -c "iptables-save -c > /etc/iptables.rules"
 /etc/network/if-pre-up.d/iptablesload Должен содержать:
#!/bin/sh
iptables-restore < /etc/iptables.rules
exit 0

Скрипт Windows
sc config MpsSvc start= auto
net start MpsSvc
net start wscsvc #запуск протоколирования Центр обеспечения безопасности
pkgmgr /iu:”TelnetClient” #enable telnet
pkgmgr /iu:”TelnetServer”
sc config TelnetServer start= Disabled
netsh advfirewall firewall add rule name="Open Port 23" dir=in action=allow protocol=TCP
localport=23 remoteip=LocalSubnet enable=yes
netsh firewall set icmpsetting 8 enable
net start TelnetServer
```

Вывод:

Было очень трудно. Научился новому. Изучил работу iptables. понял различие между OUTPUT dport и OUTPUT sport
Команды для виндовс Я просто искал. При должном желании можно не зная команд легко сконфигурировать сеть.
Возникли проблемы с запуском VirtualBox