Linux 防火墙 iptables 学习笔记（四）

1.概述

1.1 什么是 NAT

在传统的标准的 TCP/IP 通信过程中，所有的路由器仅仅是充当一个中间人的角色，也就是通常所说的存储转发，路由器并不会对转发的数据包进行修改，更为确切的说，除了将源 MAC 地址换成自己的 MAC 地址以外，路由器不会对转发的数据包做任何修改。NAT(NetworkAddressTranslation 网络地址翻译)恰恰是出于某种特殊需要而对数据包的源 ip 地址、目的 ip 地址、源端口、目的端口进行改写的操作。

1.2 为什么要进行 NAT

我们来看看再什么情况下我们需要做 NAT。

假设有一家 ISP 提供园区 Internet 接入服务，为了方便管理，该 ISP 分配给园区用户的 IP 地址都是伪 IP，但是部分用户要求建立自己的 WWW 服务器对外发布信息，这时候我们就可以通过 NAT 来提供这种服务了。我们可以再防火墙的外部网卡上绑定多个合法 IP 地址，然后通过 NAT 技术使发给其中某一个 IP 地址的包转发至内部某一用户的 WWW 服务器上，然后再将该内部 WWW 服务器响应包伪装成该合法 IP 发出的包。

再比如使用拨号上网的网吧，因为只有一个合法的 IP 地址，必须采用某种手段让其他机器也可以上网，通常是采用代理服务器的方式，但是代理服务器，尤其是应用层代理服务器，只能支持有限的协议，如果过了一段时间后又有新的服务出来，则只能等待代理服务器支持该新应用的升级版本。如果采用 NAT 来解决这个问题，

因为只在应用层以下进行处理，不但可以获得很高的访问速度，而且可以无缝的支持任何新的服务或应用。

还有一个方面的应用就是重定向，也就是当接收到一个包后，不是转发这个包，而是将其重定向到系统上的某一个应用程序。最常见的应用就是和 squid 配合使用成为透明代理，在对 http 流量进行缓存的同时，可以提供对 Internet 的无缝访问。

1.3NAT 的类型

在 linux2.4 的 NAT-HOWTO 中，作者从原理的角度将 NAT 分成了两种类型，即源 NAT(SNAT)和目的 NAT(DNAT)，顾名思义，所谓 SNAT 就是改变转发数据包的源地址，所谓 DNAT 就是改变转发数据包的目的地址。

2.原理

在“用 iptales 实现包过虑型防火墙”一文中我们说过，netfilter 是 Linux 核心中一个通用架构，它提供了一系列的"表"(tables)，每个表由若干"链"(chains)组成，而每条链中可以有一条或数条规则(rule)组成。并且系统缺省

的表是"filter"。但是在使用 NAT 的时候，我们所使用的表不再是"filter"，而是"nat"表，所以我们必须使用"-tnat"选项来显式地指明这一点。因为系统缺省的表是"filter"，所以在使用 filter 功能时，我们没有必要显式的指明"-tfilter"。

同 filter 表一样，nat 表也有三条缺省的"链"(chains)，这三条链也是规则的容器，它们分别是：

PREROUTING：可以在这里定义进行目的 NAT 的规则，因为路由器进行路由时只检查数据包的目的 ip 地址，所以为了使数据包得以正确路由，我们必须在路由之前就进行目的 NAT;

POSTROUTING：可以在这里定义进行源 NAT 的规则，系统在决定了数据包的路由以后在执行该链中的规则。

OUTPUT：定义对本地产生的数据包的目的 NAT 规则。

3.操作语法

如前所述，在使用 iptables 的 NAT 功能时，我们必须在每一条规则中使用"-tnat"显示的指明使用 nat 表。然后使用以下的选项:

3.1 对规则的操作

加入(append)一个新规则到一个链(-A)的最后。

在链内某个位置插入(insert)一个新规则(-I)，通常是插在最前面。

在链内某个位置替换(replace)一条规则(-R)。

在链内某个位置删除(delete)一条规则(-D)。

删除(delete)链内第一条规则(-D)。

3.2 指定源地址和目的地址

通过--source/--src/-s 来指定源地址(这里的/表示或者的意思，下同)，通过--destination/--dst/-s 来指定目的地址。可以使用以下四中方法来指定 ip 地址：

使用完整的域名，如“www.linuxaid.com.cn”；

使用 ip 地址，如“192.168.1.1”；

用 x.x.x.x/x.x.x.x 指定一个网络地址，如“192.168.1.0/255.255.255.0”;

用 x.x.x.x/x 指定一个网络地址，如“192.168.1.0/24”这里的 24 表明了子网掩码的有效位数，这是 UNIX 环境中通常使用的表示方法。缺省的子网掩码数是 32，也就是说指定 192.168.1.1 等效于 192.168.1.1/32。

3.3 指定网络接口

可以使用--in-interface/-i 或--out-interface/-o 来指定网络接口。从 NAT 的原理可以看出，对于 PREROUTING 链，我们只能用-i 指定进来的网络接口；而对于 POSTROUTING 和 OUTPUT 我们只能用-o 指定出去的网络接口。

3.4 指定协议及端口

可以通过--protocol/-p 选项来指定协议，如果是 udp 和 tcp 协议，还可--source-port/--sport 和--destination-port/--dport 来指明端口。

4. 准备工作

4.1 编译内核，编译时选中以下选项，具体可参看“用 iptales 实现包过虑型防火墙”一文：

```
FullNAT

MASQUERADEtargetsupport

REDIRECTtargetsupport
```

4.2 要使用 NAT 表时，必须首先载入相关模块：

```
modprobeip_tables

modprobeip_nat_ftp
```

iptable_nat 模块会在运行时自动载入。

5. 使用实例

5.1 源 NAT(SNAT)

比如，更改所有来自 192.168.1.0/24 的数据包的源 ip 地址为 1.2.3.4：

```
    iptables-tnat-APOSTROUTING-s192.168.1.0/24-oeth0-jSNAT
--to1.2.3.4
```

这里需要注意的是,系统在路由及过虑等处理直到数据包要被送出时才进行SNAT。

有一种 SNAT 的特殊情况是 ip 欺骗，也就是所谓的 Masquerading，通常建议在使用拨号上网的时候使用，或者说在合法 ip 地址不固定的情况下使用。比如

```
    #iptables-tnat-APOSTROUTING-oppp0-jMASQUERADE
```

可以看出，这时候我们没有必要显式的指定源 ip 地址等信息。

5.2 目的 SNAT(DNAT)

比如，更改所有来自 192.168.1.0/24 的数据包的目的 ip 地址为 1.2.3.4：

```
    iptables-tnat-APREROUTING-s192.168.1.0/24-ieth1-jDNAT-
-to1.2.3.4
```

这里需要注意的是，系统是先进行 DNAT，然后才进行路由及过虑等操作。

有一种 DNAT 的特殊情况是重定向，也就是所谓的 Redirection，这时候就相当于将符合条件的数据包的目的 ip 地址改为数据包进入系统时的网络接口的 ip 地址。通常是在与 squid 配置形成透明代理时使用，假设 squid 的监听端口

是 3128，我们可以通过以下语句来将来自 192.168.1.0/24，目的端口为 80 的数据包重定向到 squid 监听端口：

```
    iptables-tnat-APREROUTING-ieth1-ptcp-s192.168.1.0/24--dport80-jREDIRECT--to-port3128
```

6. 综合例子

6.1 使用拨号带动局域网上网

小型企业、网吧等多使用拨号网络上网，通常可能使用代理，但是考虑到成本、对协议的支持等因素，建议使用 ip 欺骗方式带动区域网上网。

成功升级内核后安装 iptables，然后执行以下脚本：

#载入相关模块

```
    modprobeip_tables

    modprobeip_nat_ftp
```

#进行 ip 伪装

```
    iptables-tnat-APOSTROUTING-oppp0-jMASQUERADE
```

6.2ip 映射

假设有一家 ISP 提供园区 Internet 接入服务，为了方便管理，该 ISP 分配给园区用户的 IP 地址都是伪 IP，但是部分用户要求建立自己的 WWW 服务器对外发布信息。我们可以再防火墙的外部网卡上绑定多个合法 IP 地址，然后通过 ip 映射使发给其中某一个 IP 地址的包转发至内部某一用户的 WWW 服务器上，然后再将该内部 WWW 服务器响应包伪装成该合法 IP 发出的包。

我们假设以下情景：

该 ISP 分配给 A 单位 www 服务器的 ip 为：

伪 ip：192.168.1.100

真实 ip：202.110.123.100

该 ISP 分配给 B 单位 www 服务器的 ip 为：

伪 ip：192.168.1.200

真实 ip：202.110.123.200

linux 防火墙的 ip 地址分别为：

内网接口 eth1：192.168.1.1

外网接口 eth0：202.110.123.1

然后我们将分配给 A、B 单位的真实 ip 绑定到防火墙的外网接口，以 root 权限执行以下命令：

```
ifconfigeth0add202.110.123.100netmask255.255.255.0

ifconfigeth0add202.110.123.200netmask255.255.255.0
```

成功升级内核后安装 iptables，然后执行以下脚本：

#载入相关模块

```
modprobeip_tables

modprobeip_nat_ftp
```

首先，对防火墙接收到的目的 ip 为 202.110.123.100 和 202.110.123.200 的所有数据包进行目的 NAT(DNAT)：

```
iptables-APREROUTING-ieth0-d202.110.123.100-jDNAT--to192.168.1.100

iptables-APREROUTING-ieth0-d202.110.123.200-jDNAT--to192.168.1.200
```

其次，对防火墙接收到的源 ip 地址为 192.168.1.100 和 192.168.1.200 的数据包进行源 NAT(SNAT)：

```
iptables-APOSTROUTING-oeth0-s192.168.1.100-jSNAT--to202.110.123.100

iptables-APOSTROUTING-oeth0-s192.168.1.200-jSNAT--to202.110.123.200
```

这样，所有目的 ip 为 202.110.123.100 和 202.110.123.200 的数据包都将分别被转发给 192.168.1.100 和 192.168.1.200；而所有来自 192.168.1.100 和 192.168.1.200 的数据包都将分别被伪装成由 202.110.123.100 和 202.110.123.200，从而也就实现了 ip 映射。