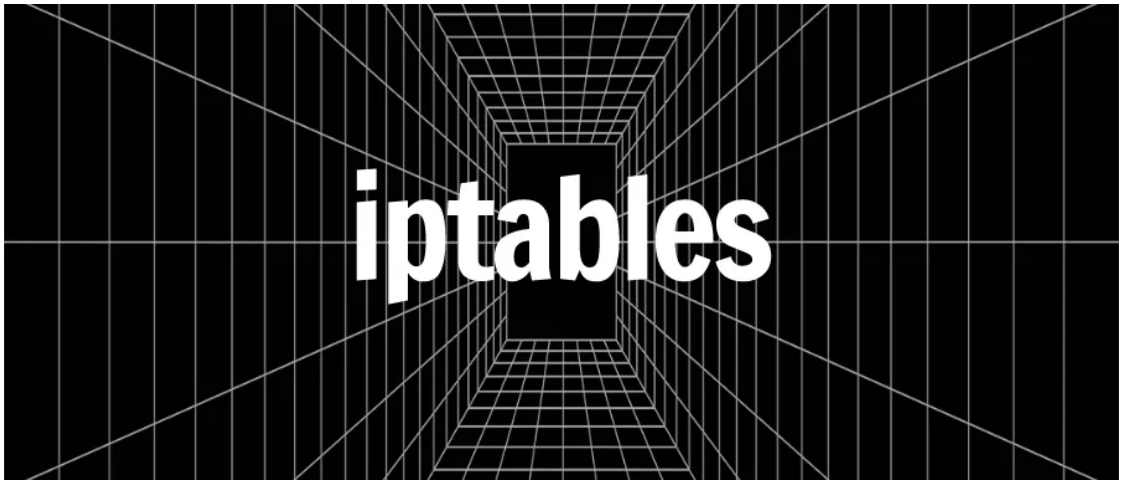
Linux下iptables 超詳細教程和使用示例

Linux公社 今天

收錄於話題
#Linux 166 #iptables 1

點擊上方藍字 ● 關注Linux公社



iptables的結構：

iptables由上而下，由Tables，Chains，Rules組成。

一、iptables的表tables與鏈chains

iptables有Filter, NAT, Mangle, Raw四種內建表：

1. Filter表

Filter是iptables的默認表，它有以下三種內建鏈(chains)：

INPUT鏈
OUTPUT鏈
FORWARD鏈

2. NAT表

NAT表有三種內建鏈：

PREROUTING鏈 它會轉換數據包中的目標IP地址（destination ip address），通常用於DNAT(destination NAT)。
POSTROUTING鏈 它會轉換數據包中的源IP地址（source ip address），通常用於SNAT（source NAT）。
OUTPUT鏈

3. Mangle表

Mangle表用於指定如何處理數據包。它能改變TCP頭中的QoS位。Mangle表具有5個內建鏈（chains）：

- PREROUTING
- OUTPUT
- FORWARD
- INPUT
- POSTROUTING

4. Raw表

Raw表用於處理異常，它具有2個內建鏈：

PREROUTING chain

OUTPUT chain

5.小結

二、IPTABLES 規則(Rules)

規則的關鍵知識點：

Rules包括一個條件和一個目標(target)
如果滿足條件，就執行目標(target)中的規則或者特定值。
如果不滿足條件，就判斷下一條Rules。

目標值（Target Values）

在target裡指定的特殊值：

ACCEPT
DROP
QUEUE
RETURN

查看各表中的規則命令

```
# iptables -t filter --list
```

查看mangle表：

```
# iptables -t mangle --list
```

查看NAT表：

```
# iptables -t nat --list
```

查看RAW表：

```
# iptables -t raw --list
```

以下例子表明在filter表的input鏈, forward鏈, output鏈中存在規則：

```
# iptables --list
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

```
Chain RH-Firewall-1-INPUT (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
2    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
3    ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
4    ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
5    ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
6    ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
7    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
8    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
9    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
10   REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```

以上輸出包含下列字段：

num – 指定鏈中的規則編號

三、清空所有iptables規則

在配置iptables之前，你通常需要用iptables --list命令或者iptables-save命令查看有無現存規則，因為有時需要刪除現有的iptables規則：

```
iptables --flush
或者
iptables -F
```

下面命令是清除iptables nat表規則。

```
iptables -t nat -F
```

四、永久生效

當你刪除、添加規則後，這些更改並不能永久生效，這些規則很有可能在系統重啟後恢復原樣。如下配置讓配置永久生效。

```
# 保存iptables规则
service iptables save

# 重启iptables服务
service iptables stop
service iptables start
```

查看當前規則：

```
cat  /etc/sysconfig/iptables
```

五、追加iptables規則

可以使用iptables -A命令追加新規則，其中 因此，一般而言，最後一條規則用於丟棄(DROP)所有數據包。如果你已經有這樣的規則了，並且使用

1.語法

```
iptables -A chain firewall-rule
```

-A chain – 指定要追加規則的鏈
firewall-rule – 具體的規則參數

2.描述規則的基本參數

以下這些規則參數用於描述數據包的協議、源地址、目的地址、允許經過的網絡接口，以及如何處理這些數據包。這些描述是對規則的基本描述。

```
-p 协议（protocol）
  指定规则的协议，如tcp, udp, icmp等，可以使用all来指定所有协议。
  如果不指定-p参数，则默认是all值。这并不明智，请总是明确指定协议名称。
  可以使用协议名(如tcp)，或者是协议值（比如6代表tcp）来指定协议。映射关系请查看/etc/protocols
  还可以使用–protocol参数代替-p参数
-s 源地址（source）
  指定数据包的源地址
  参数可以使IP地址、网络地址、主机名
  例如：-s 192.168.1.101指定IP地址
  例如：-s 192.168.1.10/24指定网络地址
  如果不指定-s参数，就代表所有地址
  还可以使用–src或者–source
-d 目的地址（destination）

```

```
  指定目的地址
  参数和-s相同
  还可以使用–dst或者–destination
-j 执行目标（jump to target）
  -j代表”jump to target”
  -j指定了当与规则(Rule)匹配时如何处理数据包
  可能的值是ACCEPT, DROP, QUEUE, RETURN
  还可以指定其他链（Chain）作为目标
-i 输入接口（input interface）
  -i代表输入接口(input interface)
  -i指定了要处理来自哪个接口的数据包
    这些数据包即将进入INPUT, FORWARD, PREROUTE链
  例如：-i eth0指定了要处理经由eth0进入的数据包
  如果不指定-i参数，那么将处理进入所有接口的数据包
  如果出现! -i eth0，那么将处理所有经由eth0以外的接口进入的数据包
  如果出现-i eth+，那么将处理所有经由eth开头的接口进入的数据包
  还可以使用–in-interface参数
-o 输出（out interface）
  -o代表”output interface”
  -o指定了数据包由哪个接口输出
  这些数据包即将进入FORWARD, OUTPUT, POSTROUTING链
  如果不指定-o选项，那么系统上的所有接口都可以作为输出接口
  如果出现! -o eth0，那么将从eth0以外的接口输出
  如果出现-i eth+，那么将仅从eth开头的接口输出
  还可以使用–out-interface参数
```

3.描述規則的擴展參數

對規則有了一個基本描述之後，有時候我們還希望指定端口、TCP標誌、ICMP類型等內容。

```
-sport 源端口（source port）针对 -p tcp 或者 -p udp
  缺省情况下，将匹配所有端口
  可以指定端口号或者端口名称，例如”-sport 22”与”-sport ssh”。
  /etc/services文件描述了上述映射关系。
  从性能上讲，使用端口号更好
  使用冒号可以匹配端口范围，如”-sport 22:100”
  还可以使用”-source-port”
--dport 目的端口（destination port）针对-p tcp 或者 -p udp
  参数和-sport类似
  还可以使用”-destination-port”
--tcp-flags TCP标志 针对-p tcp
  可以指定由逗号分隔的多个参数
  有效值可以是：SYN, ACK, FIN, RST, URG, PSH
  可以使用ALL或者NONE
--icmp-type ICMP类型 针对-p icmp
  -icmp-type 0 表示Echo Reply
  -icmp-type 8 表示Echo
```

4.追加規則的完整實例：僅允許SSH服務

本例實現的規則將僅允許SSH數據包通過本地計算機，其他一切連接（包括ping）都將被拒絕。

```
# 1.清空所有iptables规则
iptables -F

# 2.接收目标端口为22的数据包
iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT

# 3.拒绝所有其他数据包
iptables -A INPUT -j DROP
```

六、更改默認策略

上例的例子僅對接收的數據包過濾，而對於要發送出去的數據包卻沒有任何限制。本節主要介紹如何更改鏈策略，以改變鏈的行為。

1. 默認鏈策略

/!\警告 當我們使用-L選項驗證當前規則是發現，所有的鏈旁邊都有

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
```

```
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

這種情況下，如果沒有明確添加DROP規則，那麼默認情況下將採用ACCEPT策略進行過濾。除非：

```
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
iptables -A FORWARD -j DROP
```

b)更改默認策略：

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

糟糕！！如果你嚴格按照上一節的例子配置了iptables，並且現在使用的是SSH進行連接的，那麼會話恐怕已經被迫終止了！為什麼呢？因為我們已經把OUTPUT鏈策略更改為DROP了。此時雖然服務器能接收數據，但是無法發送數據：

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
DROP       all  --  anywhere             anywhere
```

```
Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
```

七、配置應用程序規則

儘管5.4節已經介紹瞭如何初步限制除SSH以外的其他連接，但是那是在鏈默認策略為ACCEPT的情況下實現的，並且沒有對輸出數據包進行限制。本節在上一節基礎上，以SSH和HTTP所使用的端口為例，教大家如何在默認鏈策略為DROP的情況下，進行防火牆設置。在這裡，我們將引進一種新的參數-m state，並檢查數據包的狀態字段。

1.SSH

```
# 1.允许接收远程主机的SSH请求
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

# 2.允许发送本地主机的SSH响应
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

- **-m state:**
- **--state: 當SSH客戶端第一個數據包到達服務器時，狀態字段為NEW；建立連接後數據包的狀態字段都是ESTABLISHED**
- **–sport 22: 因此對於SSH服務器而言，源端口就是22**
- **–dport 22: 因此對於SSH客戶端而言，目的端口就是22**

如果服務器也需要使用SSH連接其他遠程主機，則還需要增加以下配置：

```
# 1.送出的数据包目的端口为22
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

# 2.接收的数据包源端口为22
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

2.HTTP

HTTP的配置與SSH類似：

```
# 1.允许接收远程主机的HTTP请求
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

# 1.允许发送本地主机的HTTP响应
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

3.完整的配置

```
# 1.删除现有规则
iptables -F

# 2.配置默认链策略
```

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# 3.允许远程主机进行SSH连接
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 4.允许本地主机进行SSH连接
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 5.允许HTTP请求
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

配置轉發端口示例

```
iptables -t nat -I PREROUTING -p tcp --dport 3389 -j DNAT --to 38.X25.X.X02
iptables -t nat -I POSTROUTING -p tcp --dport 3389 -j MASQUERADE
```

NAT規則實戰舉例：

需求

把本地的mysql 3306端口映射出去變成63306，外面連接的語句是

```
mysql -uroot -p'password' -h xxxxx -P 63306
```

注：當訪問63306的時候，會自動去請求3306，然後返回數據。

實現

先允許數據包轉發

```
echo 1 >/proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.conf.eth0.route_localnet=1
sysctl -w net.ipv4.conf.default.route_localnet=1
```

nat規則

```
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 63306 -j DNAT --to-destination 127.0.0.1:3306
iptables -t nat -A POSTROUTING -p tcp -m tcp --dport 63306 -j SNAT --to-source 127.0.0.1
```

注：這是允許所有外來的IP訪問，慎用。

我們來做個ip限制，限制單個來源IP

```
iptables -t nat -R PREROUTING 4 -s 192.168.40.154 -p tcp -m tcp --dport 63306 -j DNAT --to-
destination 127.0.0.1:3306
iptables -t nat -R POSTROUTING 4 -s 192.168.40.154 -p tcp -m tcp --dport 63306 -j SNAT --to-source
127.0.0.1
```

注：這是只給外網的192.168.40.154連接， 其他的都連不上，

修改規則(4代表編號, --line-number可查看對應編號, -s 指定來源IP)。

查看nat規則

```
iptables -L -t nat --line-number
```

刪除nat規則

```
iptables -t nat -D POSTROUTING 1
-A 追加规则-->iptables -A INPUT
-D 删除规则-->iptables -D INPUT 1(编号)
-R 修改规则-->iptables -R INPUT 1 -s 192.168.12.0 -j DROP 取代现行规则，顺序不变(1是位置)
-I 插入规则-->iptables -I INPUT 1 --dport 80 -j ACCEPT 插入一条规则，原本位置上的规则将会往后移动一个顺位
-L 查看规则-->iptables -L INPUT 列出规则链中的所有规则
-N 新的规则-->iptables -N allowed 定义新的规则
```

關注我們

長按或掃描下面的二維碼關注

關注

每天

合作聯繫：root@linuxidc.net

喜歡此內容的人還喜歡

Linux 命令su 和sudo 的區別

杰哥的IT之旅

基於Nginx實現灰度發布與AB測試

DevOps技術棧

MySQL 跨庫分頁、分錶分頁之後，面臨的一些新問題

小哈學Java