追逐1015

25个iptables常用示例

本文将给出25个iptables常用规则示例，这些例子为您提供了些基本的模板，您可以根据特定需求对其进行修改调整以达到期望。 格式

iptables [-t 表名] 选项 [链名] [条件] [-j 控制类型] 参数

1. -P 设置默认策略:iptables -P INPUT (DROP|ACCEPT)
2. -F 清空规则链
3. -L 查看规则链
4. -A 在规则链的末尾加入新规则
5. -I num 在规则链的头部加入新规则
6. -D num 删除某一条规则
7. -s 匹配来源地址IP/MASK，加叹号"!"表示除这个IP外。
8. -d 匹配目标地址
9. -i 网卡名称 匹配从这块网卡流入的数据
10. -o 网卡名称 匹配从这块网卡流出的数据
11. -p 匹配协议,如tcp,udp,icmp
12. --dport num 匹配目标端口号
13. --sport num 匹配来源端口号

示例

1. 删除已有规则

在开始创建iptables规则之前，你也许需要删除已有规则。

1. 命令如下:
2. iptables -F
3. (or)
4. iptables –flush

2.设置链的默认策略

昵称：追逐1015
园龄：2年6个月
粉丝：0
关注：2
+加关注

< 2019年2月 >

日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Linux iptables(1)
saltstack(1)

随笔分类

docker
kubernetes
Linux(1)
mysql
openstack
python
saltstack(1)
shell

随笔档案

2017年5月 (2)

阅读排行榜

1. 25个iptables常用示例(13430)
2. saltstack安装+基本命令(1514)

推荐排行榜

1. 25个iptables常用示例(2)

链的默认政策设置为”ACCEPT”（接受），若要将INPUT,FORWARD,OUTPUT链设置成”DROP”（拒绝），命令如下：

```
1. iptables -P INPUT DROP
2. iptables -P FORWARD DROP
3. iptables -P OUTPUT DROP
```

当INPUT链和OUTPUT链都设置成DROP时，对于每一个防火墙规则，我们都应该定义两个规则。例如：一个传入另一个传出。在下面所有的例子中，由于我们已将DROP设置成INPUT链和OUTPUT链的默认策略，每种情况我们都将制定两条规则。当然，如果你相信你的内部用户,则可以省略上面的最后一行。例如：默认不丢弃所有出站的数据包。在这种情况下,对于每一个防火墙规则要求,你只需要制定一个规则——只对进站的数据包制定规则。

3. 阻止指定IP地址

例：丢弃来自IP地址x.x.x.x的包

```
1. iptables -A INPUT -s x.x.x.x -j DROP
2. 注：当你在log里发现来自某ip地址的异常记录，可以通过此命令暂时阻止该地址的访问以做更深入分析
```

例：阻止来自IP地址x.x.x.x eth0 tcp的包

```
1. iptables -A INPUT -i eth0 -s x.x.x.x -j DROP
2. iptables -A INPUT -i eth0 -p tcp -s x.x.x.x -j DROP
```

4. 允许所有SSH的连接请求

例：允许所有来自外部的SSH连接请求，即只允许进入eth0接口，并且目标端口为22的数据包

```
1. iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED
   -j ACCEPT
```

5. 仅允许来自指定网络的SSH连接请求

例：仅允许来自于192.168.100.0/24域的用户的ssh连接请求

```
1. iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 22 -m state
   --state NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED
   -j ACCEPT
```

6.允许http和https的连接请求

例：允许所有来自web - http的连接请求

```
1. iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED
   -j ACCEPT
```

例：允许所有来自web - https的连接请求

```
1. iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state
   ESTABLISHED -j ACCEPT
```

7. 使用multiport 将多个规则结合在一起

允许多个端口从外界连入，除了为每个端口都写一条独立的规则外，我们可以用multiport将其组合成一条规则。如下所示： 例：允许所有ssh,http,https的流量访问

```
1. iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state
   --state NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m
   state --state ESTABLISHED -j ACCEPT
```

8. 允许从本地发起的SSH

```
1. iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED
   -j ACCEPT
```

请注意,这与允许ssh连入的规则略有不同。本例在OUTPUT链上，我们允许NEW和ESTABLISHED状态。在INPUT链上，我们只允许ESTABLISHED状态。ssh连入的规则与之相反。

9. 仅允许从本地发起到一个指定的网络域的SSH请求

例：仅允许从内部连接到网域192.168.100.0/24

```
1. iptables -A OUTPUT -o eth0 -p tcp -d 192.168.100.0/24 --dport 22 -m state
   --state NEW,ESTABLISHED -j ACCEPT
2. iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED
   -j ACCEPT
```

10. 允许从本地发起的HTTPS连接请求

下面的规则允许输出安全的网络流量。如果你想允许用户访问互联网，这是非常有必要的。在服务器上，这些规则能让你使用wget从外部下载一些文件

```
1. iptables -A OUTPUT -o eth0 -p tcp --dport 443 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A INPUT -i eth0 -p tcp --sport 443 -m state --state ESTABLISHED
   -j ACCEPT
```

注：对于HTTP web流量的外联请求，只需要将上述命令中的端口从443改成80即可。

11. 负载平衡传入的网络流量

使用iptables可以实现传入web流量的负载均衡，我们可以传入web流量负载平衡使用iptables防火墙规则。 例：使用iptables nth将HTTPS流量负载平衡至三个不同的ip地址。

```
1. iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m
   nth --counter 0 --every 3 --packet 0 -j DNAT --to-destination
   192.168.1.101:443
2. iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m
   nth --counter 0 --every 3 --packet 1 -j DNAT --to-destination
   192.168.1.102:443
3. iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m
   nth --counter 0 --every 3 --packet 2 -j DNAT --to-destination
   192.168.1.103:443
```

12. 允许外部主机ping内部主机

```
1. iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
2. iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

13. 允许内部主机ping外部主机

```
1. iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
2. iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

14. 允许回环访问

例：在服务器上允许127.0.0.1回环访问。

```
1. iptables -A INPUT -i lo -j ACCEPT
2. iptables -A OUTPUT -o lo -j ACCEPT
```

15. 允许内部网络域外部网络的通信

防火墙服务器上的其中一个网卡连接到外部，另一个网卡连接到内部服务器，使用以下规则允许内部网络与外部网络的通信。此例中，eth1连接到外部网络(互联网)，eth0连接到内部网络(例如:192.168.1.x)。

```
1. iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

16. 允许出站的DNS连接

```
1. iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
2. iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

17. 允许NIS连接

如果你使用NIS管理用户帐户，你需要允许NIS连接。如果你不允许NIS相关的ypbind连接请求，即使SSH连接请求已被允许，用户仍然无法登录。NIS的端口是动态的，先使用命令rpcinfo –p来知道端口号，此例中为853和850端口。 rpcinfo -p | grep ypbind 例：允许来自111端口以及ypbind使用端口的连接请求

```
1. iptables -A INPUT -p tcp --dport 111 -j ACCEPT
2. iptables -A INPUT -p udp --dport 111 -j ACCEPT
3. iptables -A INPUT -p tcp --dport 853 -j ACCEPT
4. iptables -A INPUT -p udp --dport 853 -j ACCEPT
5. iptables -A INPUT -p tcp --dport 850 -j ACCEPT
6. iptables -A INPUT -p udp --dport 850 -j ACCEPT
```

注：当你重启ypbind之后端口将不同，上述命令将无效。有两种解决方案：1）使用你NIS的静态IP 2）编写shell脚本通过“rpcinfo - p”命令自动获取动态端口号,并在上述iptables规则中使用。

18. 允许来自指定网络的rsync连接请求

例：允许来自网络192.168.101.0/24的rsync连接请求

```
1. iptables -A INPUT -i eth0 -p tcp -s 192.168.101.0/24 --dport 873 -m state
   --state NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 873 -m state --state
   ESTABLISHED -j ACCEPT
```

19. 允许来自指定网络的MySQL连接请求

很多情况下，MySQL数据库与web服务跑在同一台服务器上。有时候我们仅希望DBA和开发人员从内部网络（192.168.100.0/24）直接登录数据库，可尝试以下命令：

```
1. iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 3306 -m
   state --state NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 3306 -m state --state
   ESTABLISHED -j ACCEPT
```

20. 允许Sendmail, Postfix邮件服务

Sendmail和postfix都使用了25端口，因此我们只需要允许来自25端口的连接请求即可。

```
1. iptables -A INPUT -i eth0 -p tcp --dport 25 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 25 -m state --state ESTABLISHED
   -j ACCEPT
```

21. 允许IMAP和IMAPS

例：允许IMAP/IMAP2流量，端口为143

```
1. iptables -A INPUT -i eth0 -p tcp --dport 143 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 143 -m state --state
   ESTABLISHED -j ACCEPT
```

例：允许IMAPS流量，端口为993

```
1. iptables -A INPUT -i eth0 -p tcp --dport 993 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 993 -m state --state
   ESTABLISHED -j ACCEPT
```

22. 允许POP3和POP3S

例：允许POP3访问

```
1. iptables -A INPUT -i eth0 -p tcp --dport 110 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 110 -m state --state
   ESTABLISHED -j ACCEPT
```

例：允许POP3S访问

```
1. iptables -A INPUT -i eth0 -p tcp --dport 995 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 995 -m state --state
   ESTABLISHED -j ACCEPT
```

23. 防止DoS攻击

```
1. iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-
   burst 100 -j ACCEPT
2.
3. 上述例子中：
4. -m limit：启用limit扩展
5. –limit 25/minute：允许最多每分钟25个连接（根据需求更改）。
6. –limit-burst 100：只有当连接达到limit-burst水平(此例为100)时才启用上述
   limit/minute限制。
```

24. 端口转发

例：将来自422端口的流量全部转到22端口。 这意味着我们既能通过422端口又能通过22端口进行ssh连接。启用DNAT转发。

```
1. iptables -t nat -A PREROUTING -p tcp -d 192.168.102.37 --dport 422 -j
   DNAT --to 192.168.102.37:22
```

除此之外，还需要允许连接到422端口的请求

```
1. iptables -A INPUT -i eth0 -p tcp --dport 422 -m state --state
   NEW,ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -o eth0 -p tcp --sport 422 -m state --state
   ESTABLISHED -j ACCEPT
```

25. 记录丢弃的数据表

第一步：新建名为LOGGING的链

```
1. iptables -N LOGGING
```

第二步：将所有来自INPUT链中的数据包跳转到LOGGING链中

```
1. iptables -A INPUT -j LOGGING
```

第三步：为这些包自定义个前缀，命名为”IPTables Packet Dropped”

```
1. iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables
   Packet Dropped: " --log-level 7
```

第四步：丢弃这些数据包

```
1. iptables -A LOGGING -j DROP
```

分类: Linux

标签: Linux iptables

追逐1015
关注 - 2
粉丝 - 0
+加关注

2 0

» 下一篇：saltstack安装+基本命令

posted @ 2017-05-13 01:04 追逐1015 阅读(13430) 评论(0) 编辑 收藏