ChinaUnix 讲座

iptables 高级使用研讨

cu.platinum@gmail.com

2010.08.07

最后修改时间：2010.08.07

文档维护者：白金(platinum)

v1.0.0

www.linuxidc.com

主题大纲

1. 基础部分
2. 一些误区
3. 高级技巧
4. 提高部分
5. FAQ

1. 基础部分

1.1 REDIRECT 与 DNAT 的区别？

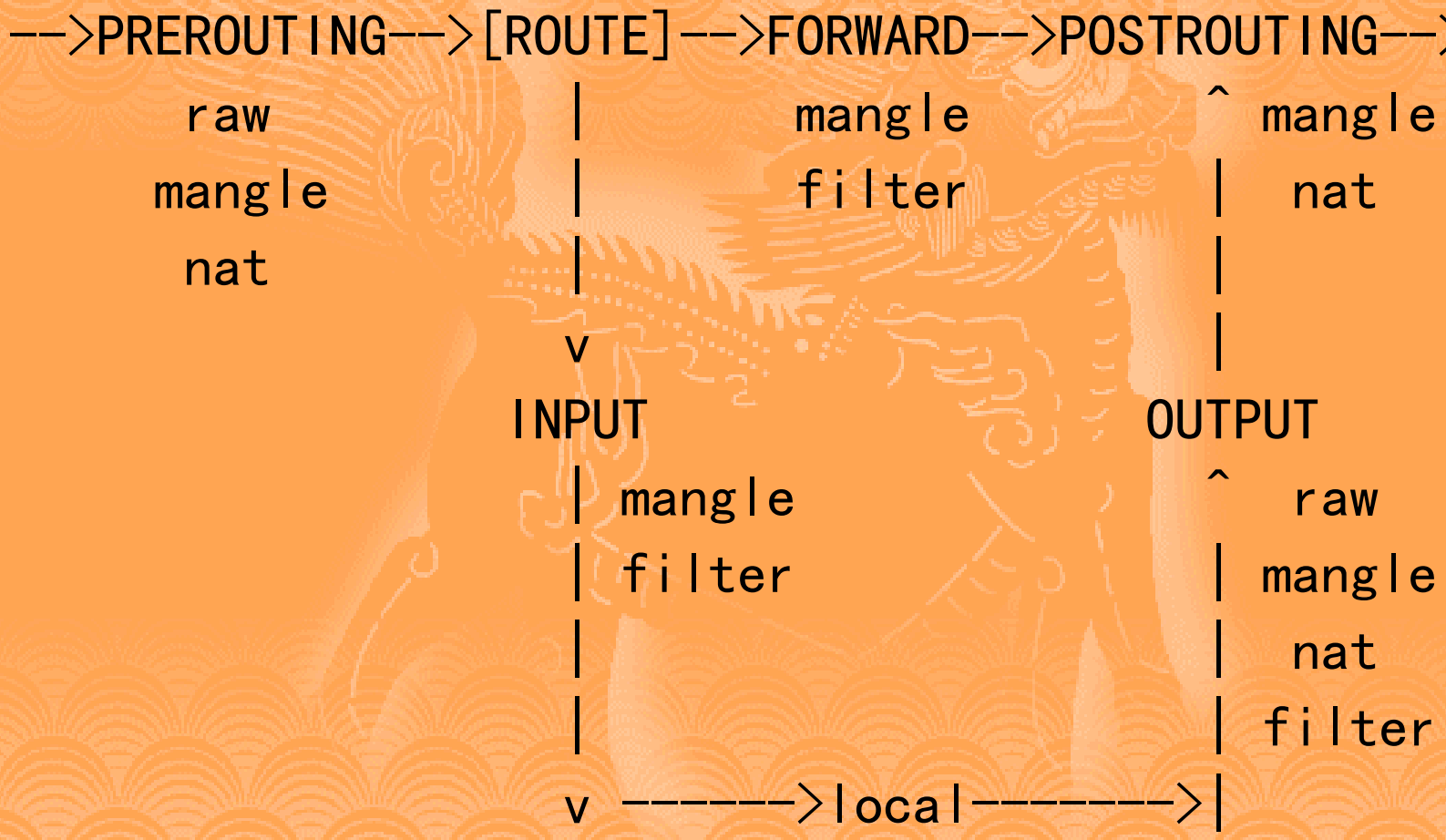
1.2 MASQUERADE 与 SNAT 的区别？

1.3 -j 与 -g 的区别？

1.4 raw 表的用途？

1.5 何为专表专用、专链专用？

1.0 回顾



1.0 回顾

语法：

```
iptables [-t table] -A <CHAIN> [rule] \
[-m <match>] [-j TARGET]
```

1.1 REDIRECT 与 DNAT 的区别？

环境：WAN(eth0), LAN(eth1)

```
iptables -t nat -A PREROUTING -i eth1 -p tcp \
--dport 80 -j REDIRECT --to 3128

iptables -t nat -A PREROUTING -i eth1 -p tcp \
--dport 80 -j DNAT --to 172.16.11.1:3128

iptables -t nat -A PREROUTING -i eth0 -p tcp \
--dport 3389 -j DNAT --to 172.16.11.250
```

结论：REDIRECT 是 DNAT 的一个特例

DNAT 的功能更强大

1.2 MASQUERADE 与 SNAT 的区别？

```
iptables -t nat -s 172.16.11.0/24 -o eth0 \
-j MASQUERADE

iptables -t nat -s 172.16.11.0/24 -o eth0 \
-j SNAT --to 123.123.123.123

iptables -t nat -s 172.16.11.0/24 -o eth0 \
-j SNAT --to 123.123.123.1-123.123.123.10
```

结论：MASQUERADE 自动根据路由选择出口

SNAT 适用于固定 IP 的环境，负载小

SNAT 可实现地址面积映射

1.3 -j 与 -g 的区别？

```
iptables -N TESTA
iptables -N TESTB
iptables -A FORWARD -s 172.16.11.1 -j TESTA
iptables -A FORWARD -s 172.16.11.2 -g TESTB
iptables -A TESTA -j MARK --set-mark 1
iptables -A TESTB -j MARK --set-mark 2
iptables -A FORWARD -m mark --mark 1 -j DROP
iptables -A FORWARD -m mark --mark 2 -j DROP
```

结论：-j(jump)相当于调用，自定义链结束后返回

-g(goto)一去不复返

1.4 raw 表的用途？

```
man iptables
iptables -t raw -vnL
iptables -t raw -A PREROUTING -i eth1 -s \
172.16.11.250 -j DROP
```

结论：raw 表工作于最前端，在 conntrack 之前

可以明确对某些数据不进行连接追踪

raw 可提前 DROP 数据，有效降低负载

1.5 何为专表专用、专链专用？

filter：专门用于过滤数据
nat：用于地址转换（只匹配初始连接数据）
mangle：用于修改数据包内容
raw：用于在连接追踪前预处理数据

PREROUTING：用于匹配最先接触到的数据(raw,mangle,nat)
INPUT：用于匹配到达本机的数据 (mangle,filter)
FORWARD：用于匹配穿越本机的数据 (mangle,filter)
OUTPUT：用于匹配从本机发出的数据
(raw,mangle,nat,filter)
POSTROUTING：用于匹配最后离开的数据 (mangle,nat)

2. 一些误区

2.1 一个公网 IP 的最大连接数?

2.2 该不该使用 LOG?

2.3 limit 到底应如何使用?

2.4 iptables 能阻止 synflood 攻击吗?

2.5 我的 iptables 脚本有什么问题?

2.1 一个公网 IP 的最大连接数？

```
static u_int32_t __hash_conntrack(const struct nf_conntrack_tuple *tuple,
                                  u16 zone, unsigned int size, unsigned int rnd)
{
        unsigned int n;
        u_int32_t h;

        /* The direction must be ignored, so we hash everything up to the
         * destination ports (which is a multiple of 4) and treat the last
         * three bytes manually.
         */
        n = (sizeof(tuple->src) + sizeof(tuple->dst.u3)) / sizeof(u32);
        h = jhash2((u32 *)tuple, n,
                   zone ^ rnd ^ (((__force __u16)tuple->dst.u.all << 16) |
                                 tuple->dst.protonum));

        return ((u64)h * size) >> 32;
}
```

结论：可以最大有 nf_conntrack_max 个连接

2.2 该不该使用 LOG?

数据包记录流程:

- 获取数据包
- 分析包信息（根据各协议计算各部分数据偏移量）
- 打印内核信息（根据不同协议打印不同信息）
- syslog 捕获内核信息
- 根据 syslog 配置决定如何输出（文件/远程）

```
iptables -A FORWARD -m state --state NEW -p tcp ! --syn \
-j LOG -log-prefix "BAD TCP "
iptables -A FORWARD -m state --state NEW -p tcp ! --syn \
-j DROP
```

结论：LOG 需占用一定负载，尽量不写硬盘

如须记录，尽量少记，否则雪上加霜

2.3 limit 到底应如何使用？

错误：

```
iptables -A FORWARD -p icmp -s 172.16.11.0/24 \
-m limit --limit 10/s -j DROP
```

正确：

```
iptables -A FORWARD -p icmp -s 172.16.11.0/24 \
-m limit --limit 10/s -j ACCEPT
iptables -A FORWARD -p icmp -s 172.16.11.0/24 \
-j DROP
```

结论：limit 仅按一定速率匹配数据

若限制，先放过一定速率数据，然后阻断

2.4 能阻止 synflood 攻击吗？

```
iptables -N syn_flood
iptables -A INPUT -p tcp --syn -j syn_flood
iptables -A syn_flood -m limit --limit 1/s -j RETURN
iptables -A syn_flood -j DROP
```

攻击强度	微弱	猛烈	无攻击
访问效果（无防）	很难访问	无法访问	正常访问
服务器负载	中	高	低
访问效果（限速）	无法访问	无法访问	极难访问
服务器负载	无	无	无

结论：限速方案是一把双刃剑，治标不治本

启用 syncookie，减小 synack_retries

增大 tcp_max_syn_backlog，其它

2.5 我的脚本有什么问题？

你的做法？

你的需求？

你的环境？

你的态度？

你的……？

结论：提问前，请参考《提问的智慧》

3. 高级技巧

3.1 如何访问部分网站时不使用 squid?

3.2 如何防止被探测 SSH 密码?

3.3 如何实现 IP/MAC 绑定?

3.4 如何防止小路由（及如何破解）?

3.5 如何防止被 tracert?

3.6 如何实现服务器负载分担?

3.7 如何得知内网用户流量?

3.8 如何对 DNS 域名进行过滤?

3.9 如何有效阻止疯狂下载人士?

3.1 访问部分网站不使用 squid?

```
iptables -t nat -A PREROUTING -i eth1 -s \
172.16.11.250 -j ACCEPT
iptables -t nat -A PREROUTING -i eth1 -d \
123.123.123.123 -j ACCEPT
iptables -t nat -A PREROUTING -i eth1 -s \
172.16.11.0/24 -j REDIRECT --to 3128
```

结论：让无需转向的数据提前脱离“魔掌”

3.2 如何防止被探测 SSH 密码？

```
iptables -A INPUT -p tcp --dport 22 \
-m state --state NEW -m recent --set \
--name SSH --rsource -m recent --name SSH \
--update --seconds 10 --hitcount 4 \
--rsource -j DROP
```

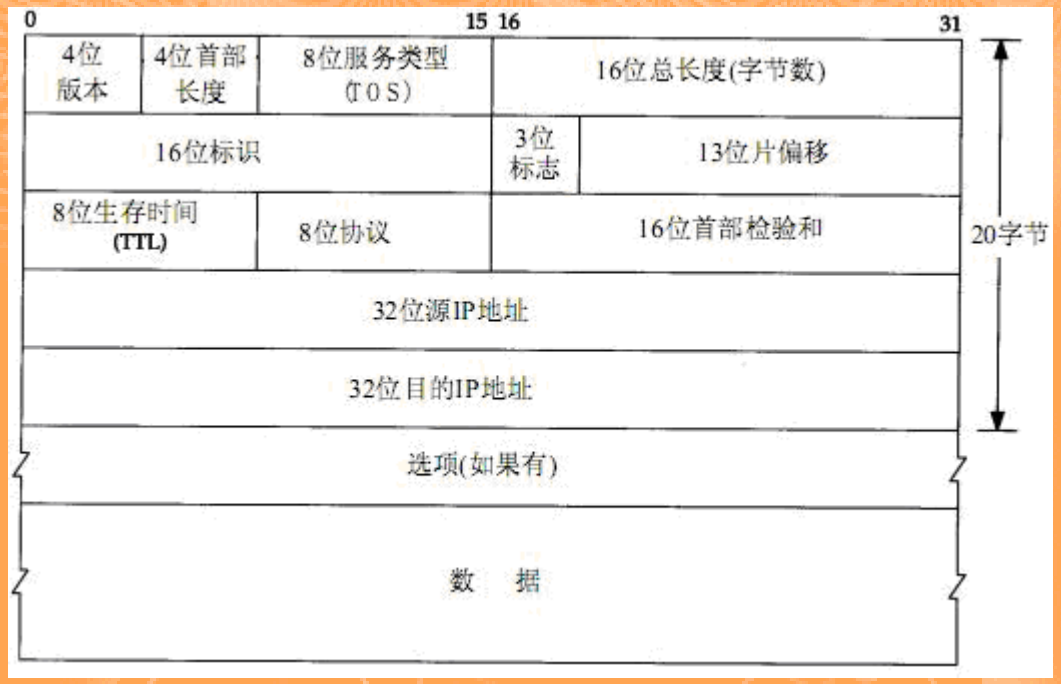
结论：模块可以在同一个规则内反复使用

优点：合并使用可降低负载

3.3 如何实现 IP/MAC 绑定？

```
iptables -N MAC_CHECK
iptables -A FORWARD -i eth1 -o eth0 -j MAC_CHECK
iptables -A MAC_CHECK -s 172.16.11.101 -m mac \
--mac-source XX:XX:XX:XX:XX:XX -j RETURN
iptables -A MAC_CHECK -s 172.16.11.102 -m mac \
--mac-source YY:YY:YY:YY:YY:YY -j RETURN
iptables -A MAC_CHECK -s 172.16.11.103 -m mac \
--mac-source ZZ:ZZ:ZZ:ZZ:ZZ:ZZ -j RETURN
iptables -A MAC_CHECK -j DROP
```

3.4 如何防止小路由（及破解）？




```
iptables -N TTL_CHECK
iptables -A FORWARD -i eth1 -o eth0 -j TTL_CHECK
iptables -A TTL_CHECK -m ttl --ttl-eq 128 -j RETURN
iptables -A TTL_CHECK -m ttl --ttl-eq 64 -j RETURN
iptables -A TTL_CHECK -m ttl --ttl-eq 255 -j RETURN
iptables -A TTL_CHECK -j DROP
```

3.5 如何防止被 tracert?

ping ?= tracert

ping = ICMP

tracert = TTL 试探

```
iptables -A INPUT -m ttl --ttl-eq 1 -j DROP
iptables -A INPUT -m ttl --ttl-lt 4 -j DROP
iptables -A FORWARD -m ttl --ttl-lt 6 -j DROP
```

如何一跳直达?

3.6 如何实现服务器负载分担？

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-m statistic --mode nth --every 3 -j DNAT --to 172.16.11.101
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-m statistic --mode nth --every 2 -j DNAT --to 172.16.11.102
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j DNAT --to 172.16.11.103
```

statistic 的其它用途？

```
iptables -A FORWARD -i eth1 -o eth0 -s 172.16.11.250 -m \
statistic --mode random --probability 0.1 -j DROP
```

3.7 如何得知内网用户流量？

http://www.intra2net.com/en/developer/ipt_ACCOUNT/index.php

```
iptables -A FORWARD -j ACCOUNT --addr \
172.16.11.0/24 --tname LOCALNET

iptaccount -a
iptaccount -l LOCALNET
iptaccount -l LOCALNET -f
```

crond + shell + ACCOUNT

3.8 如何对 DNS 域名进行过滤？

为什么 string 模块无法匹配？

DNS 数据包结构？

string 和 layer7 的实现：

```
iptables -A FORWARD -m string --algo bm --hex-string \
"|03|www|09|chinaunix|03|com" -j DROP
```

```
dns_CU
\x03www\x09chinaunix\x03net
iptables -A FORWARD -m layer7 --l7proto dns_CU -j DROP
```

3.9 如何有效阻止疯狂下载人士？

connlimit：限制每个用户的连接数
connbytes：限制长连接数据速率
quota：设置每个用户流量配额
statistic：设置用户随机丢包量

4. 提高部分

4.1 如何提高 iptables 脚本的性能？

4.2 如何对 conntrack 进行性能调优？

4.3 如何对 layer7 的匹配功能进行加强？

4.1 如何提高脚本的性能？

MASQUERADE/SNAT

规则数量

规则顺序

匹配顺序

巧用 raw 表提前处理

少做/不做 LOG

4.2 如何对 conntrack 进行调优？

```
modprobe nf_conntrack hashsize=200000
modprobe nf_conntrack_ipv4
sysctl -w net.netfilter.nf_conn \
track_max=500000
```

注意：需要大内存，用空间换时间

4.3 如何对 layer7 进行加强？

```
} else if(!strcmp(info->protocol, "unset")) {
        pattern_result = 2;
        DPRINTK("layer7: matched unset: not yet classified "
                "(%d/%d packets)\n",
                total_acct_packets(master_conntrack), num_packets);
/* Put your code here */
} else if (!strcmp(info->protocol, "platinum") &&
           match_platinum(skb)) {
        pattern_result = 1;
/* If the regexp failed to compile, don't bother running it */
} else if(comppattern &&
          regexec(comppattern, master_conntrack->layer7.app_data)){
        DPRINTK("layer7: matched %s\n", info->protocol);
        pattern_result = 1;
} else pattern_result = 0;
```

5. FAQ

谢 谢