Haproxy 使用 tproxy 实现透明代理

实验环境

Server1 为代理服务器，有两个网卡

eth0:192.168.1.55　用于对外服务

eth1:10.0.0.1　gw 10.0.0.254 用于内网通讯

Server2 为应用服务器，一块网卡

eth0:10.0.0.2　gw 10.0.0.1

service 的网关一定要配成 service 的内网 IP

准备工作

1. 检查系统内科是否已支持 tproxy

```
[root@localhost /]# grep TPROXY /boot/config-`uname -r`
CONFIG_NETFILTER_TPROXY=m
CONFIG_NETFILTER_XT_TARGET_TPROXY=m
[root@localhost /]#
```

2. 安装 haproxy

编译参数

make TARGET=linux26　USE_LINUX_TPROXY=1

make install PREFIX=/usr/local/haproyx

安装完成后，检查 haproxy 是否支持 tproxy

```
[root@localhost /]# /usr/local/haproxy/sbin/haproxy -vv | grep OPTIONS
  OPTIONS = USE_LINUX_TPROXY=1 USE_STATIC_PCRE=1
[root@localhost /]#
```

Haproxy.cfg

```
global
        daemon
        stats socket /var/run/haproxy.stat mode 600
        log 127.0.0.1 local4
        maxconn 40000
        ulimit-n 80013
        pidfile /var/run/haproxy.pid
defaults
        log global
        mode       http
        contimeout          4000
        clitimeout          42000
        srvtimeout          43000
        balance roundrobin
listen    VIP-222
        bind      192.168.1.222:80
        mode       http
        option    forwardfor
        source 0.0.0.0 usesrc clientip
        cookie    SERVERID insert nocache indirect
        server server1 10.0.0.2:80 weight 1 cookie server1 check
```

server backup 127.0.0.1:80 backup
option redispatch

在 server1 上配置网络、iptables 及内核参数（可加入/etc/rc.local）
net.ipv4.ip_forward = 1
net.ipv4.conf.all.send_redirects = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.forwarding = 1

```
iptables -t mangle -N DIVERT
#在 mangle 中新建一条规则链 DIVERT
iptables -t mangle -A PREROUTING -p tcp -m socket -j DIVERT
#凡进入 PREROUTING 链的报文都跳转去新定义的 DIVERT 链
iptables -t mangle -A DIVERT -j MARK --set-mark 1
#凡进入 DIVERT 链的报文均使用 mangle 表的标记功能标记上 1 标记
iptables -t mangle -A DIVERT -j ACCEPT
#进入 DIVERT 链的报文均导入路由
ip rule add fwmark 1 lookup 100
#添加 100 策略路由表,并将由 iptables 打了标记 1 的数据报文从 100 路由表规定路由流动
可以在/etc/iproute2/rt_tables 查看或者使用此命令查看 ip rule sh 本机的所有路由表
ip route add local 0.0.0.0/0 dev lo table 100
#由于到达本地的数据包要想成功到达，必须要找到一条 local 路由(类型对即可，无需非要
在 local 表)，因此为路由表 100 确定默认路由，进入环流
```

最后启动相关软件即可

Linux 路由查找流程
在 Linux 中，内置了三张路由表：
local，main，default，其中 local 路由表的优先级最高，并且不能被替换，在有数据包进来的时候，首先无条件的查找 local 路由表，如果找到了路由，则数据包就是发往本机的，如果找不到，则接着在其它的路由表中进行查找。使用 ip route ls table local 命令可以看到 local 表的内容，比如机器的 eth0 网卡上配有 192.168.0.7，则在 local 表中会有如下的路由：
local 192.168.0.7 dev eth0 proto kernel scope host src 192.168.0.7
local 表中的路由是可以删除的。路由的 src 项指的是当数据包从本机发出时，在 local 表中找到了源地址的路由，首选的源 ip 地址
在 local 表和 main 表之间，可以插入 251 张策略路由表，因此如果有策略路由表的话，如果 local 表中没有找到路由，则会查找策略路由表

Iptables 规则表之间的优先顺序
当数据包抵达防火墙时，将依次应用 raw、mangle、nat 和 filter 表中对应链内的规则
mangle 表，包含五个规则链：PREROUTING、POSTROUTING、INPUT、OUTPUT、FORWARD
mangle 表主要用于修改数据包的 TOS（服务类型）、TTL（生存周期）值以及为数据包设置 Mark 标记，以实现 Qos（服务器质量）调整以及策略路由等应用
Iptables 规则链之间的优先顺序

入站数据流向：来自外界的数据包到达防火墙后，首先被 PREROUTING 规则链处理（是否修改数据包地址等），之后进行路由选择（判断该数据包应发往何处），如果数据包的目标地址是防火墙本机（如 Internet 用户访问防火墙主机中 web 服务端数据包），那么内核将其传递给 INPUT 链进行处理(决定是否允许通过等),通过以后在交给系统上层的应用程序(如 httpd 服务器）进行响应
PREROUTING：在对数据包做路由选择之前，应用此链中的规则

转发数据流向：来自外界的数据包到达防火墙后，首先被 PREROUTING 规则链处理，之后进行路由选择,如果数据包的地址是其外部地址(如果局域网通过网关访问 QQ 站点的数据包)，则内核将其传递给 FORWARD 链进行处理（是否转发或拦截），然后再交给 POSTROUTING 规则链（是否修改数据包地址等）进行处理

出站数据流向：防火墙本机向外部地址发送到数据包（如在防火墙主机中测试公网 DNS 服务时），首先被 OUTPUT 规则链处理，之后进行路由选择，然后传递给 POSTROUTING 规则链（是否修改数据包的地址等）进行处理

iptables 规则链内部各条防火墙规则之间的优先顺序
在数据包经由每条规则的处理过程中，依次按第一条、第二条.....的顺序进行匹配，如果找到一条能够匹配该数据包的规则，则不继续检查后边的规则，如果比对完整个规则链,也找到和数据包相匹配的规则，就按照规则链的默认策略进行处理

零下七度：411898301