IPTABLES

IPCafe.net

Definitions

IPTables is a widely used firewall tool that interfaces with the linux kernel's netfilter packet filtering framework.

Netfilter or IPTables

There may be some confusion about the difference between Netfilter and iptables. Netfilter is an infrastructure; it is the basic API that the Linux 2.4 kernel offers for applications that want to view and manipulate network packets. Iptables is an interface that uses Netfilter to classify and act on packets.

Tables and Chains

IPTable has 3 built-in tables and each of them has its own chains:

-FILTER:the filter table is used to make decisions about whether to let packet continue to its intended destination or to deny its request
Chains:INPUT,FORWARD,OUTPUT

-NAT:The nat table is used for address translation
Chain:PREROUTING, OUTPUT, POSTROUTING

-MANGLE:The mangle table is used to alter the IP headers of the packet in various ways.
Chain:PREROUTING,INPUT,FORWARD,OUTPUT,POSTROUTING

*The default table is filter

Rule Actions

A firewall rule specifies criteria for a packet and a target. If the packet does not match, the next rule in the chain is the examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain or one of the special values:ACCEPT,DROP,QUEUEorRETURN

ACCEPT:means to let the packet through.

DROP:means to drop the packet on the floor

QUEUE:means to pass the packet to userspace

RETURN:means stop traversing this chain and resume at the next rule in the previous (calling)chain

Configuration

By default ip forwarding is disables in iptables.

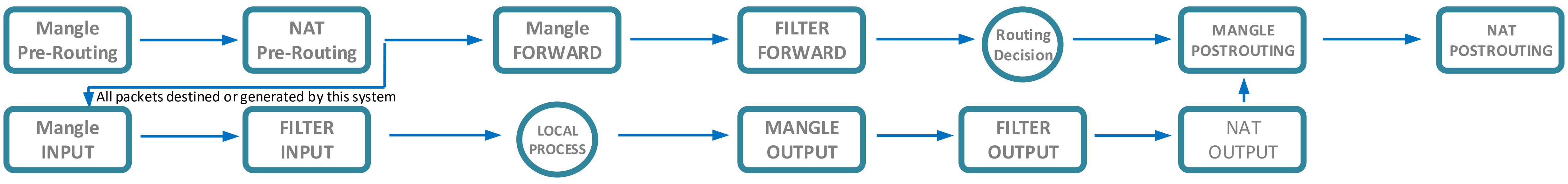
To enable ip forwarding : echo 1 > /proc/sys/net/ipv4/ip_forward then service iptables restart

Iptables –F	*Delete all rules in iptables*
Iptables –t nat -F	*Delete all rules in nat table*
Iptables –t mangle -F	*Delete all rules in mangle table*
Iptables –t nat –D [rule number]	*Delete specific rule number in mangle table*
Iptables –L –v –n	*Show ruls in iptables*
Service iptables save	*Saves configurations*
Iptables –A INPUT –i eth1 –j LOG	*Enables log for interface Ethernet1*
Iptables –A INPUT –i eth1 –s 10.0.0.0/8 –j LOG	*Enables log for interface Ethernet1 with source 10.0.0.0/8*
Tail –f /var/log/messages	*Viewing logs*

Examples

Deny Port 80	*Iptables –A INPUT –p tcp --dport 80 –J DROP* *Iptables –A INPUT –i eth1 –p tcp --dport 80 –J DROP*
Allow private network access to internet(eth0 connected to internet)	*Iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE*
Drop ssh port for specific IP	*Iptables –A INPUT –s 172.17.100.100 –p tcp --dport 22 –j DROP*
URL filtering for ipcafe.net	*Iptables –A FORWARD –m string --string "ipcafe.net" --algo kmp --to 65535 –j DROP*
Disable URL filtering for ipcafe.net	*Iptables –D FORWARD –m string --string "ipcafe.net" --algo kmp --to 65535 –j DROP*
Map public ip to private ip	*Iptables –t nat –A PREROUTING –d 5.39.1.1 –j DNAT --to-destination 172.17.100.1*
Map public ip:port to private ip:port	*Iptables –t nat –A PREROUTING –d 5.39.1.1 --dport 8080 –j DNAT --to-destination 172.17.100.1:1625*
Mark packet and then DNAT	*Iptables –t mangle –A PREROUTING –s 172.16.100.1 –p icmp --icmp-type 8 –j MARK --set-mark 777* *Iptables –t nat –A PREROUTING –m mark --mark 777 –j DNAT --to-destination 172.16.100.1*
Permit echo-reply and echo-request	*Iptables –A INPUT –p icmp --icmp-type 8 –s 0/0 –d 172.16.100.1 –m state --state NEW,ESTABLISHED,RELATED –j ACCEPT* *Iptables –A OUTPUT –p icmp --icmp-type 0 –s 172.16.100.1 –d 0/0 –m state --state NEW,ESTABLISHED,RELATED –j ACCEPT*

IPTables chain order



Mangle Pre-Routing → NAT Pre-Routing → Mangle FORWARD → FILTER FORWARD → Routing Decision → MANGLE POSTROUTING → NAT POSTROUTING

All packets destined or generated by this system

Mangle INPUT → FILTER INPUT → LOCAL PROCESS → MANGLE OUTPUT → FILTER OUTPUT → NAT OUTPUT → MANGLE POSTROUTING

Chains available within each Table

MANGLE

Pre-Routing

All packets entering the system in any way, before routing decides whether the packet is to be forwarded or is destined locally (INPUT chain).

Usecase: "mark" on the packet for further processing in other tables-use in PBR-NAT

Post-Routing

All packets leaving the system go through this chain.

Usecase: mangling on packets before they leave our host,but after the actual routing decisions.

Input

All packets destined for this system go through this chain.

usecase: At this point, the mangle INPUT chain is hit.We use this chain to mangle packets, after they have been routed, but before they are actually sent to the process on the machine.

Forward

All packets merely passing through the system go through this chain.

usecase: All packets created by this system go through this chain.

Output

All packets created by this system go through this chain.

usecase:This can be used for very specific needs, where we want to mangle the packets after the initial routing decision, but before the last routing decision made just before the packet is sent out.

NAT

Pre-Routing

Incoming packets pass through this chain before the local routing table is consulted, primarily for DNAT.

Usecase: change destination port and IP.

Post-Routing

Outgoing packets pass through this chain after the routing decision has been made, primarily for SNAT.

Usecase: This chain should first and fore most be used for SNAT.

Output

Allows limited DNAT on locally-generated packets.

Usecase: This chain can be used to NAT outgoing packets from the firewall itself.

FILTER

Pre-Routing

All packets destined for this system go through this chain

Usecase: This is where we do filtering for all incoming traffic destined for our local host. Note that all incoming packets destined for this host pass through this chain

Forward

All packets merely passing through the system (beingrouted) go through this chain.

Usecase: All packets merely passing through the system(being routed) go through this chain.

Output

All packets created by this system go through this chain.

Usecase: This is where we filter packets going out from the local host.

By Seyed Hamidreza Hoseini

Version:0.9