Ipchains and Iptables

Linux operating system natively supports packet-filtering rules:

- Kernel versions 2.2 and earlier support the ipchains command.
- Kernel version 2.4 support the iptables command.
- ipchains and iptables are mutually exclusive.

Ipchains

The ipchains application uses three built-in chains, often called *special chains*. They are as follows:

- **input** Used to control packets entering the interface.
- **output** Used to control packets leaving the interface.
- **forward** Used to control packets being masqueraded, or sent to remote hosts.

Ipchains (2)

You must specify a target using the -j option.

- Allowed target built-in values are ACCEPT, DENY, REJECT, MASQUERADE, REDIRECT, and RETURN.
- The MASQUERADE target allows you to establish NAT on a firewall.
- Case is important for both the chains and the targets.
  - all chains are in lowercase letters, and
  - all targets are in uppercase

Examples of Ipchains

```
ipchains -A input -p icmp -s 0/0 -d 0/0 -j REJECT
```

- This command tells the input chain to forbid any ICMP traffic from any host

```
ipchains -A input -p icmp –s 10.100.100.0/24 -d 0/0 -j REJECT
```

- This command blocks ICMP traffic from only the 10.100.100.0/24 network
- The host can no longer receive packets, but it can still send them

```
ipchains -A output -p icmp -s 192.168.2.0/24 –d 10.100.100.0/24 -j REJECT
```

- Prohibits this host from sending packets

Examples of Ipchains (2)

- You are not, of course, limited to controlling just ICMP traffic. If you want to block incoming POP3 traffic from all hosts, you issue the following command:

```
ipchains -A input -p tcp -s 0/0 -d 0/0 110 -j REJECT
```

- If you want to deny all traffic by default and then specifically allow only, say, POP3 traffic, you could use the -P option, which sets a policy for the chain you specify.
- You could then begin to allow the POP3 traffic, as well the DNS service and
- the ephemeral ports necessary for your system to connect to a POP3 server:

Ipchains Policy option -P

- ipchains -P output DENY
- ipchains -P forward DENY
- ipchains -P input DENY
- ipchains -A input -p tcp -s 0/0 -d 0/0 110 -j ACCEPT
- ipchains -A input -p tcp -s 0/0 -d 0/0 1024: -j ACCEPT
- ipchains -A input -p udp -s 0/0 -d 0/0 1024: -j ACCEPT
- ipchains -A output -p tcp -s 0/0 -d 0/0 1024: -j ACCEPT
- ipchains -A output -p udp -s 0/0 -d 0/0 1024:-j ACCEPT
- ipchains -A output -p udp -s 0/0 -d 0/0 53 -j ACCEPT

IP Masquerade

- The following entry would enable all systems that are using the internal NIC as a default gateway to use the Internet:
- ipchains -A forward -i eth0 –s 192.168.2.0/24
  -d 0/0 -j MASQ
- The above entry adds an entry to the forward chain, which is designed to allow masquerading

Iptables

iptables keeps the filter table and adds filter and nat tables.

- **filter** Contains the INPUT, OUTPUT, and FORWARD chains. This is the default table, and it will report its contents when you list chains using the iptables -L command.
- **nat** Used for creating NAT tables. Contains the PREROUTING, OUTPUT,and POSTROUTING tables. The PREROUTING table alters packets as soon as they enter (used when masquerading connections), the OUTPUT table alters locally generated packets, and POSTROUTING alters packets before they are about to be sent on the network.
- **mangle** Alters the packets. Generally, you do not use this for establishing NAT.

Examples of Iptables

To create a simple personal firewall that blocks all incoming ICMP traffic, you issue the following command:

- iptables -A INPUT -p icmp -s 0/0 -d 0/0 -j DROP

To block ICMP traffic from only the 10.100.100.0/24 network, you issue this command:

- iptables -A INPUT -p icmp -s 10.100.100.0/24 -d 0/0 -j DROP

Blocking all Incoming ICMP

- To deny all but POP3 traffic, you issue the following commands, after flushing any existing rules:
- iptables -P INPUT DROP
- iptables -P FORWARD DROP
- iptables -P OUTPUT DROP
- iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 1024: -j ACCEPT
- iptables -A INPUT -p udp -s 0/0 -d 0/0 --dport 1024: -j ACCEPT
- iptables -A OUTPUT -p udp -s 0/0 -d 0/0 --dport 53 -j ACCEPT
- iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 --dport 110 -j ACCEPT

IP Masquerade

If you want to masquerade a connection using iptables, you would use the nat table. Using the same scenario as the ipchains command, you would masquerade your internal network so that it could connect to the Internet as follows:

- iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

Stateful Iptables Firewall

If you are using a laptop or a desktop computer you may want to prevent any attempt from the outside world to establish a connection with your computer

- iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

You can allow all outgoing connections originating from your computer with the following command

- iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT