Iptables Quick Tutorial

Wang Xiaolin
wx672ster@gmail.com

March 11, 2018

What’s A Packet Filter?

A packet filter is a piece of software which looks at the `header` of packets as they pass through, and decides the fate of the entire packet. It might decide to

- `DROP` the packet (i.e., discard the packet as if it had never received it),
- `ACCEPT` the packet (i.e., let the packet go through), or
- something more complicated.

Why Packet Filtering?

Control — allow certain types of traffic, and disallow others.

Security — you might not want outsiders telnetting to your Linux box.

Watchfulness — It's nice to tell the packet filter to let you know if anything abnormal occurs.

Packet Filter Under Linux

iptables talks to the kernel and tells it what packets to filter.

The iptables tool inserts/deletes rules from the kernel's packet filtering table.

Quick Start

Debian/Ubuntu users can do:

```
stud@debian:~$ sudo apt-get install iptables
stud@debian:~$
stud@debian:~$ sudo iptables -A INPUT -s 147.8.212.123 -p all -j DROP
stud@debian:~$
stud@debian:~$ sudo iptables -D INPUT -s 147.8.212.123 -p all -j DROP
stud@debian:~$
stud@debian:~$ man iptables
stud@debian:~$
stud@debian:~$ ls /usr/share/doc/iptables/html
stud@debian:~$
```

Terminology

Filter table is in the kernel, contains `chains`.

Chains a.k.a. firewall chains, are lists of filtering rules. The three kernel built-in chains are called `INPUT`, `OUTPUT`, and `FORWARD`.

Rules Each rule says:

```
  if the packet header looks like this
then here’s what to do with the packet
```

How Chains Work?

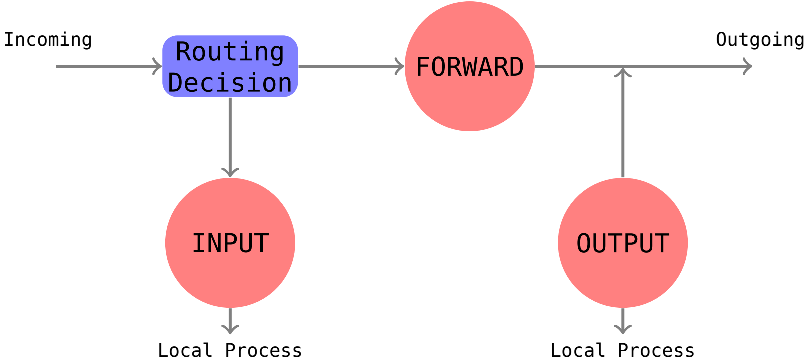

Figure: Chains

Using iptables

To manage whole chains:

1. Create a new chain (`-N`).
2. Delete an empty chain (`-X`).
3. Change the policy for a built-in chain. (`-P`).
4. List the rules in a chain (`-L`).
5. Flush the rules out of a chain (`-F`).
6. Zero the packet and byte counters on all rules in a chain (`-Z`).

To manipulate rules inside a chain:

1. Append a new rule to a chain (`-A`).
2. Insert a new rule at some position in a chain (`-I`).
3. Replace a rule at some position in a chain (`-R`).
4. Delete a rule at some position in a chain, or the first that matches (`-D`).

Examples

```
stud@debian:~$ ping -c 1 127.0.0.1
stud@debian:~$
stud@debian:~$ sudo iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
stud@debian:~$
stud@debian:~$ ping -c 1 127.0.0.1
stud@debian:~$
stud@debian:~$ sudo iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
stud@debian:~$
stud@debian:~$ sudo iptables -A INPUT -s ! 127.0.0.1 -p all -j DROP
stud@debian:~$
stud@debian:~$ sudo iptables -A INPUT -s 192.168.1.0/24 -p all -j DROP
stud@debian:~$
```

More Examples

```
~$ # Syn-flood protection:
~$ sudo iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
~$
~$ # Furtive port scanner:
~$ sudo iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT
~$
~$ # Ping of death:
~$ sudo iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
~$
```

References I

- P. Srisuresh, M. Holdrege, IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663 (Informational), Internet Engineering Task Force, **1999-08**.

- P. Srisuresh, K. Egevang, Traditional IP Network Address Translator (Traditional NAT), RFC 3022 (Informational), Internet Engineering Task Force, **2001-01**.

- P. Ferguson, D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827 (Best Current Practice), Updated by RFC 3704, Internet Engineering Task Force, **2000-05**.

- G. Ziemba, D. Reed, P. Traina, Security Considerations for IP Fragment Filtering, RFC 1858 (Informational), Updated by RFC 3128, Internet Engineering Task Force, **1995-10**.

References II

- I. Miller, Protection Against a Variant of the Tiny Fragment Attack (RFC 1858), RFC 3128 (Informational), Internet Engineering Task Force, **2001-06**.
- W. contributors, Iptables — Wikipedia, The Free Encyclopedia, [Online; accessed 11-March-2018], **2017**.
- T. Bautts, T. Dawson, G. Purdy, *Linux Network Administrator's Guide*, O'Reilly Media, **2005**.
- C. Hunt, *TCP/IP Network Administration*, O'Reilly Media, **2002**.