Linux Networking: IP Packet Filter Firewalling

David Morgan

© David Morgan 2003,2004

Firewall types

- Packet filter
- Proxy server

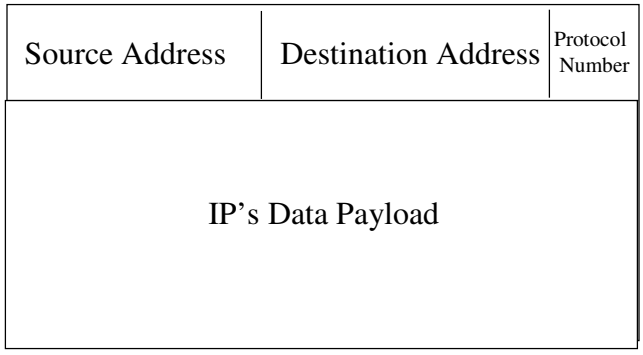
© David Morgan 2003,2004

Linux “Netfilter” Firewalling

- Packet filter, not proxy
- Centerpiece command: iptables
- Starting point: packet structure details

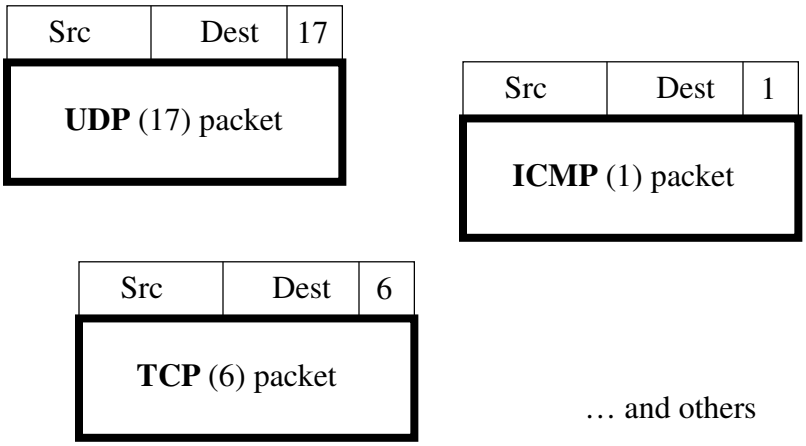
© David Morgan 2003,2004

IP packet structure



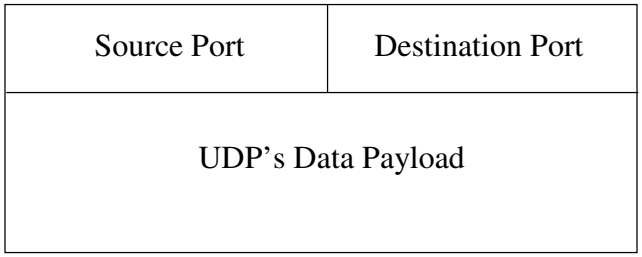
© David Morgan 2003,2004

Payload types - subprotocols



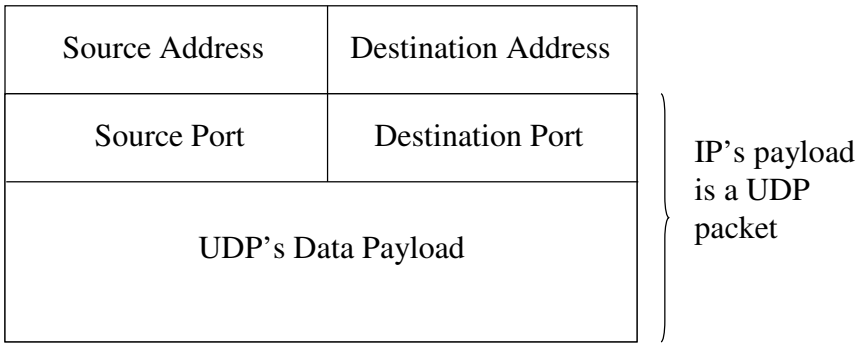
© David Morgan 2003,2004

UDP packet structure



© David Morgan 2003,2004

UDP/IP packet structure



© David Morgan 2003,2004

Address? Port?

- Address number
  - **a <u>Machine</u>** designator
  - identifies one among multiple machines on a network
    - 198.186.203.55 identifies linux.com on internet
    - 198.137.241.43 identifies whitehouse.gov
- Port number
  - **a <u>Task</u>** designator
  - identifies one among multiple tasks in a machine
    - 80 identifies web server running on linux.com
    - 22 identifies secure shell server running on linux.com

© David Morgan 2003,2004

Address? Port?

Two address-port pairs uniquely define a process to process "conversation" across a network

Source Address	Destination Address
Source Port	Destination Port

e.g., when President Bush browses linux.com:

198.137.241.43 (whitehouse.gov)	198.186.203.55 (linux.com)
62102	80

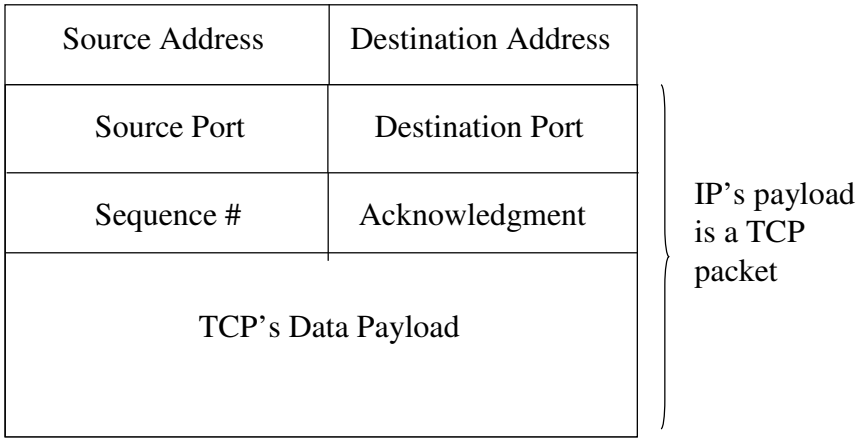
© David Morgan 2003,2004

TCP packet structure

Source Port	Destination Port
Sequence #	Acknowledgment
TCP's Data Payload	

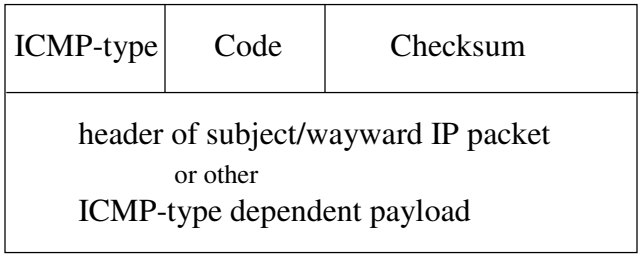
© David Morgan 2003,2004

TCP/IP packet structure



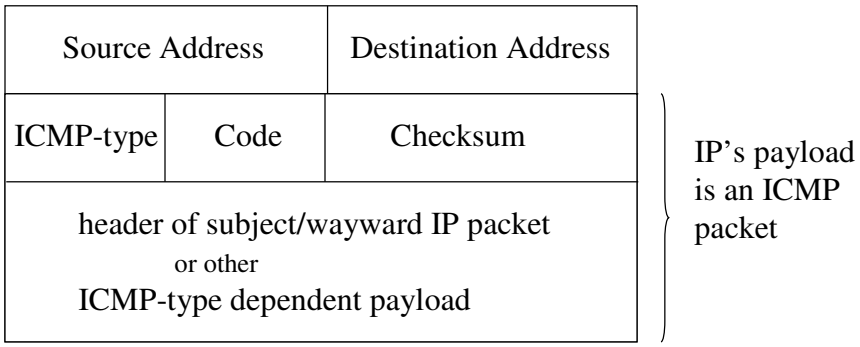
© David Morgan 2003,2004

ICMP packet structure



© David Morgan 2003,2004

ICMP/IP packet structure



© David Morgan 2003,2004

UDP packet example

Used for internet name service inquiries

```
[root@EMACH1 /root]# ping www.acme.com
```

63370	53
"www.acme.com's IP address please"	

© David Morgan 2003,2004

TCP packet example

Used for webserver-to-webbrowser traffic

80	1050
Sequence #	Acknowledgment
website text & images	

© David Morgan 2003,2004

ICMP packet example

Used for echo (ping) requests…

```
[root@EMACH1 /root]# ping www.acme.com
```

8	Code	Checksum
echo request – “are you there?”		

© David Morgan 2003,2004

ICMP packet example

… and for echo (ping) relies.

0	Code	Checksum
echo reply – “yes I’m here”		

© David Morgan 2003,2004

Firewall = ruleset

- An in-memory datastructure by whose elements packets that appear at interfaces are evaluated
- A corresponding series of commands, each invocation of which populates the table with a single element
- Elements are called “rules”

© David Morgan 2003,2004

Firewall - iptables

- low level - iptables command (to compose individual rule)
- middle level - Ziegler firewall (a particular ruleset)
- high level – firewall.local (custom extensions to firewall)

© David Morgan 2003,2004

Iptables organization

- Tables (have chains)
  - filter table
  - nat table
- Chains (contain rules)
  - filter
    - INPUT chain
    - OUTPUT
    - FORWARD
  - nat
    - PREROUTING chain
    - POSTROUTING

© David Morgan 2003,2004

An Individual Rule

- Condition - Examines and qualifies a packet
- Action - Operates on the packet if it qualifies

© David Morgan 2003,2004

What a Rule says

- "If a packet's header looks like this, then here's what to do with the packet"
- "looks like this" e.g.
  - goes to a certain (range of) address(es) or
  - uses the telnet port, 23 or
  - is an ICMP packet
- "what to do" e.g.
  - pass it
  - discard it

© David Morgan 2003,2004

```
iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535
 -s 192.168.4.0/24 -d 0.0.0.0/0 –j ACCEPT
```

– Table for this rule

– Rule action
  - -A add rule to chain/list
  - -D delete rule from chain/list
  - -P default policy for chain/list

– Rule chain/list (tables contain chains)
  - INPUT
  - OUTPUT
  - FORWARD
  - PREROUTING
  - POSTROUTING

– Packet qualifiers
  - By interface and direction
  - protocol
  - source port number(s)
  - destination port number(s)
  - source address (range)
  - destination address (range)

– Packet disposition
  - ACCEPT
  - DROP
  - REJECT
  - SNAT
  - DNAT

© David Morgan 2003,2004

What a Chain is

- ordered checklist of regulatory rules
  – Multiple rules, for packets with particular characteristics
  – Single rule for default (catch-all) policy

- operation
  – Packet tested against rules in succession
    - First matching rule determines “what to do” to packet
  – If packet matches no rule
    - Chain’s default policy determines “what to do” to packet

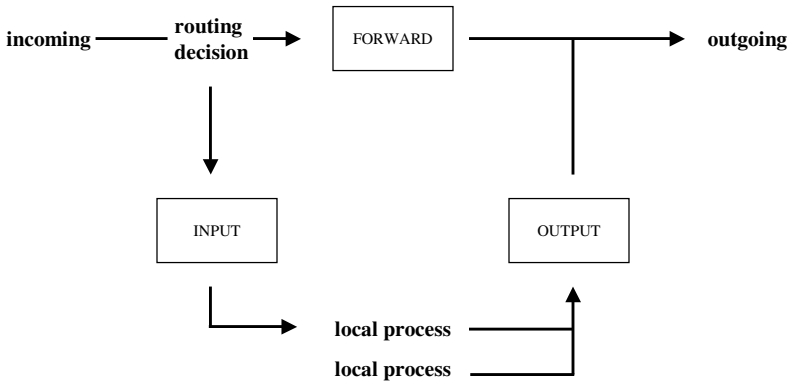
© David Morgan 2003,2004

Multiple chains

- Input chain
  - When arriving at an interface, do we let a packet come in?
- Output chain
  - When departing from an interface, do we let a packet go out?
- Forwarding chain
  - When traversing this machine to another, do we let a packet pass between interfaces?

© David Morgan 2003,2004

Filter traversal by packets



© David Morgan 2003,2004

A 4-rule filtering firewall

```
iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23
 -s 0.0.0.0/0 -d 192.168.4.1/32 –j ACCEPT

iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535
 -s 192.168.4.1/32 -d 0.0.0.0/0 –j ACCEPT

iptables -t filter -P INPUT DROP

iptables -t filter -P OUTPUT DROP
```

Executed in chronological sequence as shown, resultant 2-chain firewall permits telnet access between this machine 192.168.4.1 and others via eth1. And nothing else.

© David Morgan 2003,2004

Priority of chronology = priority of effect

```
iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23
 -s 64.1.1.1/32 -d 192.168.4.1/32 –j DROP

iptables -t filter -A INPUT -i eth1 -p tcp --sport 1024:65535 --dport 23
 -s 0.0.0.0/0 -d 192.168.4.1/32 –j ACCEPT

iptables -t filter -A OUTPUT -o eth1 -p tcp --sport 23 --dport 1024:65535
 -s 192.168.4.1/32 -d 0.0.0.0/0 –j ACCEPT

iptables -t filter -P INPUT DROP

iptables -t filter -P OUTPUT DROP
```

… EXCEPT no telnet from machine 64.1.1.1, because first rule eclipses second since it preceded it. (Second not reached, never applied.)

© David Morgan 2003,2004

nat table: rules that alter packet

- Masquerading

```
iptables -t nat -A POSTROUTING
        -o eth1 -s 10.0.0.0/8
                -j SNAT --to 216.83.185.193
```

- Pinholing (port forwarding)

```
iptables -t nat -A PREROUTING
        -i eth1 -d 216.83.185.193/32 -p tcp --dport 5631
                -j DNAT --to 216.83.185.193
```

© David Morgan 2003,2004

Firewall ruleset philosophies

- Optimistic/lax
  - set everything open
  - apply selective closures
- Pessimistic/strict
  - set everything closed
  - apply selective openings

© David Morgan 2003,2004

Set everyting closed

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

© David Morgan 2003,2004

Firewall persistence

- firewall is memory-resident
- volatile across reboot
- re-create at bootime by init script containing
  - individual iptables commands (e.g., Zeigler), or
  - iptables-restore and iptables-save (e.g., RedHat)

© David Morgan 2003,2004

Init script basics

- UNIX has a conventional method to uniformly start/stop services
- one script per service in /etc/rc.d/init.d
- scripts accept parameters start, stop, or restart
- if firewall's script is:
  /etc/rc.d/init.d/firewall
- start it with:
  /etc/rc.d/init.d/firewall start, or
  service firewall start

© David Morgan 2003,2004

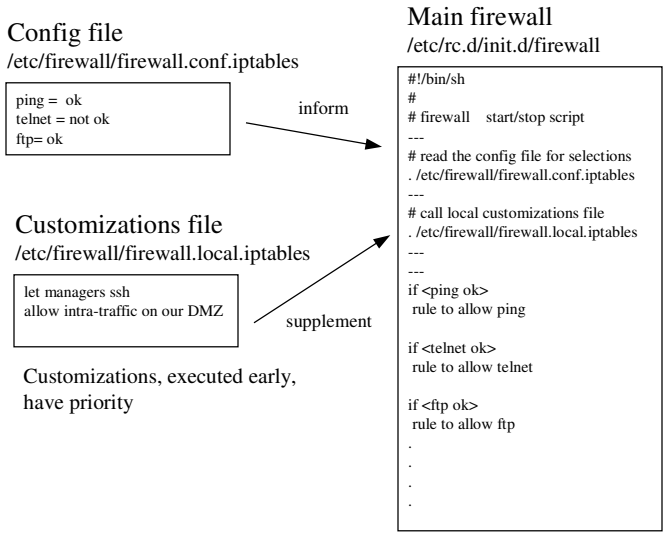
Zeigler-style firewall

- Zeigler firewall in /etc/rc.d/init.d/firewall
- configuration file in /etc/firewall/firewall.conf
- firewall script full of individual iptables invocations (i.e., rules)
- config files set variables for conditional evaluation in main firewall script

© David Morgan 2003,2004

Zeigler Firewall

- files & organization



© David Morgan 2003,2004

RedHat-default-style firewall

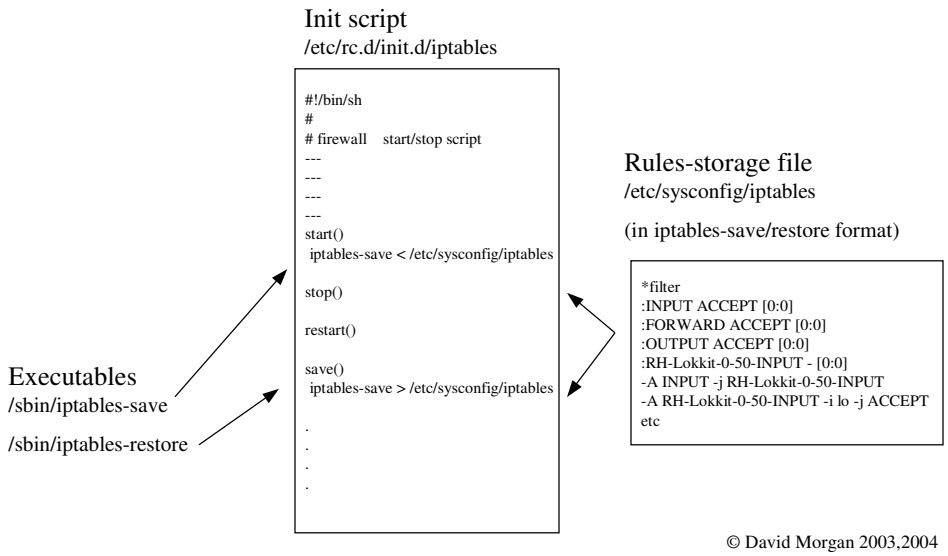
- firewall stored whole in /etc/sysconfig/iptables
- utility pair: iptables-restore, iptables-save
- use their own (ascii) format
- installtime high/medium/low security write different /etc/sysconfig/iptables
- init script /etc/rc.d/init.d/iptables iptables-restores from /etc/sysconfig/iptables
- modified firewall can be stored for future by

  service iptables save

© David Morgan 2003,2004

RedHat Firewall

- files & organization



© David Morgan 2003,2004

Please see …

http://www.iptables.org/

Linux Firewalls, 2nd edition, Robert Zeigler, New Riders, 2002

http://linux-firewall-tools.com/ (Zeigler's site)

http://www.malibyte.net/iptables/scripts/fwscripts.html (3rd-party derivative of Ziegler fw)

© David Morgan 2003,2004