ipfw & IP Filter

Yung-Zen Lai (yzlai@tp.edu.tw)
2004/10
Agenda

- Network and Firewall Basics
- `ipfw` – FreeBSD IP Firewall and Traffic Shaper
  - Firewall
  - Traffic Shaper
- IP Filter – TCP/IP Firewall/NAT Software
  - Firewall
  - Network Address Translation
Network Basics – OSI 7 Layer

Network architecture based on the OSI model

Layer
7  Application  Application Protocol  Application

4  Transport

3  Network

2  Data Link

1  Physical

Exchange Unit
APDU

Transport Protocol

Communication Subnet Boundary

Network

Network

Network

Network

Data Link

Data Link

Data Link

Data Link

Frame

Packet

TPDU

Bit

Physical

Physical

Physical

Physical
Network Basics – TCP, UDP

TCP
- Transmission Control Protocol
- Connection-Oriented
- Three-Way handshake

UDP
- User Datagram Protocol
- Connectionless
Firewall Basics

- filter packets based on their components
  - IP
  - TCP
  - UDP
  - ICMP
  - Other Protocol

- to perform an action on packets that match the filter.
  - Pass
  - Reject
  - Discard
  - Log
ipfw – Firewall

- ipfw
  - options IPFIREWALL in KERNCONF
  - first match
  - add/delete
  - list/show
  - flush
  - zero/resetlog
  - set disable/enable
  - set move/swap/show
ipfw Firewall rules

n ipfw add / delete [ number ]
  q < allow | reject | deny | reset | unreach >
  q < all | icmp | tcp | udp >

n from
  q < src-net/host > < src-port >

n to
  q < dst-net/host > < dst-port >

n [ < via interface > / < options > ]
ipfw Firewall rules (cont.)

- **Allow packets that match rule**
  - `allow | accept | pass | permit`
  - `ipfw add allow all from me to any`
  - `ipfw add allow all from smtp.tp.edu.tw to me`

- **Discard packets that match this rule.**
  - `reject | deny | drop`
  - `ipfw add deny all from any to 224.0.0.0/8`

- **Send some notice back**
  - `reset(TCP), unreach <code>(ICMP)`
  - `ipfw add reset tcp from any to any 23`
ipfw Firewall rules – Rule Actions

- `< allow | accept | pass | permit >`
- `< check-state >`
- `< count >`
- `< divert > port`
- `< fwd | forward > ipaddr[, port]`
- `< reject >`
- `< reset >`
- `< skipto > number`
- `< tee > port`
- `< unreach > code`
ipfw Firewall rules – Rule Body

* ip from \{ x or not y or z \} to any
* [proto from src to dst] [options]
* src and dst: \{ addr | \{ addr or ... \} \} [[not] ports]
* addr: [not] \{ any | me | addr-list | addr-set \}
  * any
  * matches any IP address.
  * me
  * matches any IP address configured on an interface in the system.
ipfw Firewall rules – Rule Body

n  ip-addr:
   q  numeric-ip | hostname
   q  addr/masklen
   q  addr:mask

n  ports: {port | port-port}[,ports]
ipfw Firewall rules – Rule Options

- established
- frag
- icmptypes types
- in | out
- keep-state
- limit {src-addr | src-port | dst-addr | dst-port} N
- setup
# Telnet/SSH access control (controlled by hosts.allow)
- ipfw add pass tcp from any to me 22 setup
- ipfw add pass tcp from any to me 23 setup

# Allow setup of SMTP/POP3
- ipfw add pass tcp from any to me 25 setup
- ipfw add pass tcp from any to me 110 setup

# Allow setup of DNS
- ipfw add pass tcp from any to me 53 setup
- ipfw add pass udp from any to me 53

# Default to deny
- ipfw add add 65500 reset log tcp from any to any
ipfw Firewall rules – Dynamic Rules

ipfw add check-state
ipfw add deny tcp from any to any established
ipfw add allow tcp from my-net to any setup keep-state
ipfw add allow tcp from my-net/24 to any setup limit src-addr 10
ipfw add allow tcp from any to me setup limit src-addr 4
ipfw Firewall sets of rules

- 32 different sets, numbered 0 to 31
- Set 31 is reserved for the default rule
- Rules are put in set 0 by default
- # Allow icmp (ping and traceroute only)
  - `ipfw add set 1 pass icmp from any to any icmptypes 0,3,8,11`
  - `ipfw add set 1 pass udp from any to any 33434-34000`

- `ipfw set move {rule rule-number | old-set} to new-set`
- `ipfw set swap first-set second-set`
**ipfw – Traffic Shaper**

- options `DUMMYNET` in `KERNCONF`

- `ipfw`
  - `<pipe> number config pipe-configuration`
  - `<queue> number config queue-configuration`

- Parameters can be configured for a pipe
  - `bw [bandwidth | device]`
    - `ipfw pipe 1 config bw 300Kbit/s`
  - `delay ms-delay`

- Parameters can be configured for a queue
  - `pipe pipe_nr`
  - `weight weight`
ipfw – Traffic Shaper examples

- Limit traffic from local clients on 192.168.2.0/24
  - ipfw add pipe 1 ip from 192.168.2.0/24 to any out
  - ipfw pipe 1 config bw 300Kbit/s queue 50KBytes

- Simulate a bidirectional link with bandwidth limitations
  - ipfw add pipe 1 ip from any to any out
  - ipfw add pipe 2 ip from any to any in
  - ipfw pipe 1 config bw 64Kbit/s queue 10Kbytes
  - ipfw pipe 2 config bw 64Kbit/s queue 10Kbytes
ipfw – Traffic Shaper examples

- introduce some delay in the communication
  - ipfw add pipe 1 ip from any to any out
  - ipfw add pipe 2 ip from any to any in
  - ipfw pipe 1 config delay 250ms bw 1Mbit/s
  - ipfw pipe 2 config delay 250ms bw 1Mbit/s
IP Filter – Firewall

ipfilter
  q  options IPFILTER in KERNCONF
  q  Last/first match
  q  ipf -F <a|i&o|s|S> -f <filename>
  q  ipfstat -i/-o
  q  ipnat –C/-F/-l/-s

The official IPF homepage
ipfilter Firewall rules

pass | block | nomatch

in | out

[ log ]

[ quick ]

[ proto < tcp | udp | icmp > ]

from

< src-net / host / all > [ port = XX | icmp-type X ]

to

< dst-net / host / all > [ port = XX | icmp-type X ]
ipfilter Firewall rules (cont.)

n [ on < interface > ]

n [ options ]
  q [ flags < flag > ]
  q keep state
  q keep frags
Responding To a Blocked Packet

- block return-rst in ...
  - Return RST packet in TCP

- block return-icmp(port-unr) in ...
  - Return port-unreachable in ICMP using firewall’s IP address

- block return-icmp-as-dest(port-unr) in ...
  - Return port-unreachable in ICMP using destination’s IP address

Fancy Logging Techniques

- block in log level auth.info quick ...
- block in log level auth.alert quick ...
ipfilter Firewall rules (cont.)

n **Allow packets that match rule**
  q pass out from 163.21.249.172 to any
  q pass in from smtp.tp.edu.tw to any

n **Discard packets that match this rule.**
  q block in proto udp from any to any port = 137
  q block in proto udp from any to any port = 138

n **Send some notice back**
  q block return-rst in proto tcp all flags S
  q block return-icmp[return-code]
# Telnet/SSH access control (controlled by hosts.allow)
- pass in proto tcp from any to any port = ssh flags S keep state
- pass in proto tcp from any to any port = telnet flags S keep state

# Default to deny
- block in log all
- block return-rst in proto tcp all flags S
Rule Groups

- [head X] & [group X]

  block out quick on bge0 all head 1
  block out quick on bge1 all head 2
  block in quick from 192.168.0.0/16 to any group 1
  block in quick from 172.16.0.0/12 to any group 1
  pass out quick proto tcp from any to
     163.21.249.128/25 port = 80 flags S keep state
group 1
ipnat – ipfilter NAT

/etc/rc.conf
  gateway_enable="YES"
  ipnat_enable="YES"
  ipnat_rules="/etc/ipnat.rules"

ipnat.rules
  map bge0 192.168.100.0/24 -> 163.21.249.172/32
  map bge0 192.168.100.0/24 -> 163.21.249.172/32 portmap tcp/udp 40000:60000
  map bge0 192.168.100.0/24 -> 163.21.249.172/32 proxy port ftp ftp/tcp
Mapping Many Addresses Into One Address

- map bge0 192.168.100.0/24 -> 163.21.249.172/32
- map bge0 192.168.100.0/24 -> 0/32
- map bge0 192.168.100.0/24 -> 163.21.249.172/32 portmap tcp/udp 40000:60000
  - transport tcp/udp into the port range of 40000 to 60000
- map bge0 192.168.100.0/24 -> 163.21.249.172/32 portmap tcp/udp auto
- map bge0 192.168.100.0/24 -> 163.21.249.172/32 proxy port ftp ftp/tcp

Application Proxies
Mapping Many Addresses Into a Pool of Addresses

\[
\text{map bge0 192.168.100.0/24 -> 163.21.249.192/26}
\]

Policy NAT

\[
\text{map bge0 from 192.168.100.0/24 ! to 140.122.0.0/16 -> 163.21.249.191/32}
\]

One to One Mappings

\[
\text{bimap bge0 192.168.100.1/32 -> 163.21.249.190/32}
\]
ipnat (cont.)

- **Spoofing Services**
  - `rdr bge0 163.21.249.172/32 port 80 -> 192.168.0.5 port 80`

- **Transparent Proxy Support**
  - `rdr bge0 0.0.0.0/0 port 80 -> 127.0.0.1 port 3128`

- **Using NAT As a Load Balancer**
  - `rdr bge0 163.21.249.172/32 port 80 -> 192.168.0.5 port 80 tcp round-robin`
  - `rdr bge0 163.21.249.172/32 port 80 -> 192.168.0.6 port 80 tcp round-robin`
The End

Thank you!