Bridge Firewall Configuration


Temasys lee
temasys@yahoo.co.kr
Introduction

The Scenario
   - Bridge Configuration
   - PF Firewall Rules Configuration

Bridge Testbed
Introduction

Bridge Firewall

- Connection between Hub (Switch Hub) and IP network using Layer 2 Frame protocol.
  - Switch Hub and Bridge.

- Bridge Firewall using OpenBSD Box Nic 2 NIC to filter Non-IP traffic through Bridge rules packet passing traffic.
  - OpenBSD Kernel level filter
  - OpenBSD Non-IP NIC filtering IP traffic.
  - L2, L3, L4 layer traffic.
The Scenario

Requirements

Performance PF OpenBSD

- 350Mhz or Higher x86 class processor
- 256MB of RAM
- 4GB hard drive
- 2 NIC (High quality recommended, e.g., Intel, 3com)
  - Management IP NIC 1 NIC
- OpenBSD 3.3 or Higher
  - QoS (Default QoS)

[root@root]# uname - amnprsv
OpenBSD 3.3 vpn#4 i386 Intel Pentium II ("GenuineIntel" 686-class, 512KB L2 cache)
[root@root]# dmesg
OpenBSD 3.3- stable (vpn) #4: Thu Nov  6 17:12:36 PST 2003
   root@openbsd.openable.net:/usr/src/sys/arch/i386/compile/vpn
cpu0: Intel Pentium II ("GenuineIntel" 686-class, 512KB L2 cache) 351 MHz
................................
The Scenario (cont_#2)

- Bridge Configuration
  - Setting up the bridge
    - OpenBSD installation is finished, remove any IP address information.
      - `man bridge`
    - We need to enable ip forwarding between the two network interface.
      - `edit your /etc/sysctl.conf:
        net.inet.ip.forwarding=1        # 1=Permit forwarding (routing) of packets`
    - We Need to enable pf firewall. (edit /etc/rc.conf)
      - `edit your /etc/rc.conf
        pf=YES                # Packet filter / NAT
        pf_rules=/etc/pf.conf             # Packet filter rules file`
    - The 2 bridge interfaces are fxp0 and fxp1. (If your interface names are different, change the interface names accordingly)
      - `# echo "up" > /etc/hostname.fxp0   (External interface)
        # echo "up" > /etc/hostname.fxp1   (Internal interface)
        # echo "add xl0 add xl1 up" > /etc/bridgename.bridge0`
The Scenario (cont_#3)

Bridge Configuration (cont_#2)

- Verity the bridge is up by running.
  - You should see output that includes this entry.
  - `# ifconfig -a`

```
[root@root]# ifconfig -a
fxp0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    address: 00:90:27:34:48:d3
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active

fxp1: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    address: 00:a0:c9:8b:8e:51
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active

fxp2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    address: 00:03:47:b0:5b:17
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    inet 100.100.11.2 netmask 0xffffff00 broadcast 100.100.11.255

pflog0: flags=141<UP, RUNNING, PROMISC> mtu 33224

bridge0: flags=41<UP, RUNNING> mtu 1500
```

........
The Scenario (cont_#4)

PF Firewall Rules Configuration

- pf(Packet Filter) is the packet filtering system in OpenBSD 3.0 and later.
- Its syntax and functionality are very similar to ipf in both FreeBSD, NetBSD, and earlier versions of OpenBSD. If you are familiar with ipf ruleset syntax, pf syntax should be readable.

Traffic problem <Importance>

- When using state keeping on a bridge, the packet goes through PF twice; it is an incoming packet on one interface, and an outgoing packet on the other.

  - Solutions
    - Ext_if(fxp0) , Int_if(fxp1) = < State keep >
      - Ext_if Rules → Allow all traffic traversing Ext_if.
      - Int_if Rules → Rules of the pf assign for a task.
The Scenario (cont_#5)

PF Firewall Rules Configuration (cont_2)

Simple Firewall Rules

##### outside interface
pass in on fxp0 from any to any keep state
pass out on fxp0 from any to any keep state

##### internal interface
pass in on fxp1 from any to any keep state
pass out on fxp1 from any to any keep state
The Scenario (cont_#6)

Advanced Firewall Ruleset.

Our example Filtering requirements:

- All ssh traffic will be allowed from the internet to any machine on the network.
- All http traffic will be allowed form the internet to our webserver.
- All UDP domain(for DNS lookups) and ntp will be allowed in.
- ICMP echo request/reply (ping) will be allowed.
- All access out from our network to the Internet will be allowed.
- We want to keep state on all inbound connections.
- We want to keep state on all outbound connections.
- We want to log all dropped packets.
The Scenario (cont_#7)

Advanced Firewall Ruleset(cont_2)

#       $OpenBSD: pf.conf,v 1.19 2003/03/24 01:47:28 ian Exp $
ext_if="fxp0"
int_if="fxp1"
man_if="fxp2"

##### Management Interface 100.100.11.2 - - - > 100.100.11.1
pass in quick on fxp2 all
pass out quick on fxp2 all

#####External Bridge interface rules ((allow all in - filter on internal)
# In bridge mode, We only filter on one interface.
pass in quick on $ext_if all
pass out quick on $ext_if all

#Block and Log everything In by default
block out log on $int_if all - - - > �� �� �� �� Drop.
The Scenario (cont_#8)
Advanced Firewall Ruleset(cont_3)

#######Incoming Packet Rules

- Allowed incoming tcp services (ssh, telnet, http, domain, ntp)
  - pass out on $int_if proto tcp from any to any port = 22 keep state
  - pass out on $int_if proto tcp from any to any port = 23 keep state
  - pass out on $int_if proto tcp from any to 192.168.135.248 port = http keep state
  - pass out on $int_if proto udp from any to any port { domain, ntp } keep state
  - pass out on $int_if proto tcp from any to 192.168.135.248 port 21 keep state

- Allow ICMP (ping) IN
  - pass out/in certain ICMP queries and keep state (ping)
  - pass out on $int_if inet proto icmp all icmp- type 8 code 0 keep state

#######OUT Rules

- pass in on $int_if inet proto icmp all icmp- type 8 code 0 keep state
- Pass (Allow) all UDP/TCP Out and keep state
  - pass in on $int_if proto udp all keep state
  - pass in on $int_if proto tcp all modulate state

Edited by temasys.. 2003/12/11/14:29
The Scenario (cont_#8)

- pf commands
  - flush current filter rules & reload:
    - `# /sbin/pfctl -F rules && /sbin/pfctl -R /etc/pf.conf`
  - flush current nat rules & reload:
    - `# /sbin/pfctl -F nat && /sbin/pfctl -N /etc/nat.conf`
  - show filter information (statistics and counters):
    - `# pfctl -s info`
  - to display the current list of active MAP/redirect filters and active sessions:
    - `# /sbin/pfctl -s state`
  - to find out the ``hit'' statistic for each individual rule in /etc/pf.conf:
    - `# /sbin/pfctl -s rules -v`
  - watch port scans going by on the screen:
    - `/var/log/pflog` is a binary file generated by pflogd so you can't just view it. Use tcpdump instead:
      - `# tcpdump -i pflog0 -r /var/log/pflog`
❖ **Bridge Testbed**

❖ **TestBed**
- OpenBSD® Bridge PF firewall
- QoS traffic

- Routing
  - pf Stateful packet Inspection
  - Routing Firewall
- Bridge PFQoS.

- Bridge firewall.
Q n A