Practical use of OpenBSD routing domains with redundant firewalls

Matthieu Herrb

EuroBSDCon - 16-17 september 2023

https://homepages.laas.fr/matthieu/talks/pf-rdomain.pdf

Licence

About the author

Research Engineer at CNRS-LAAS in Toulouse, France.

- PhD in Robotics, 1991
- Support for robotics projects (Linux, Embedded/Realtime systems)
- Security for the whole lab → OpenBSD Firewalls

And also

- OpenBSD contributor (Xenocara) since 1997
- Member of Tetaneutral.net, a non-profit ISP + Hosting association

Mastodon:
@mherrb@tetaneutral.net
@matthieu@bsd.network

The LAAS network context

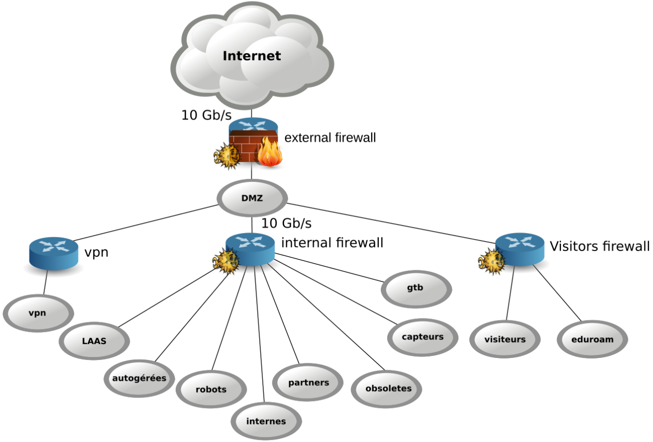
LAAS is a CNRS laboratory with:

- around 650 staff members
- around 100 PhD defenses / year
- 6 research departments from Computer Science to Nano Technologies and Robotics
- a set of hosted servers (storage, virtualization, computing cluster, backups,...)
- around 2000 workstations (Windows, Linux, macOS,...) and experimental devices connected to the network
- some security sensitive research projects
- OpenBSD firewalls in production since 2009 (JRES 2011 presentation)
- almost fully IPv6 enabled since 2004

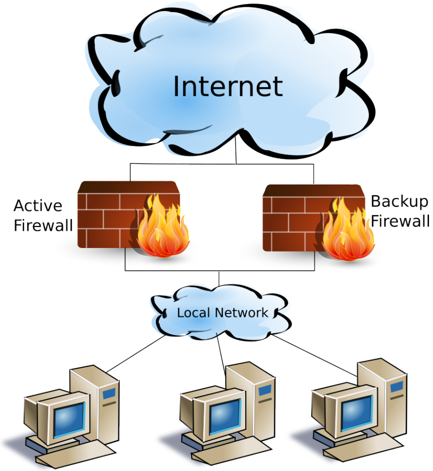
Other uses of OpenBSD at LAAS

- DNS authoritative servers (isc-bind)
- Caching / recursive DNS servers (unbound + carp)
- SMTP servers
- NTP servers
- DHCP servers
- Web proxy (squid)
- arpwatch probe

Network architecture



Redundant firewalls with Carp



The backup firewall problem

In the “classic” Carp setup the backup (passive) firewall is hard to manage :

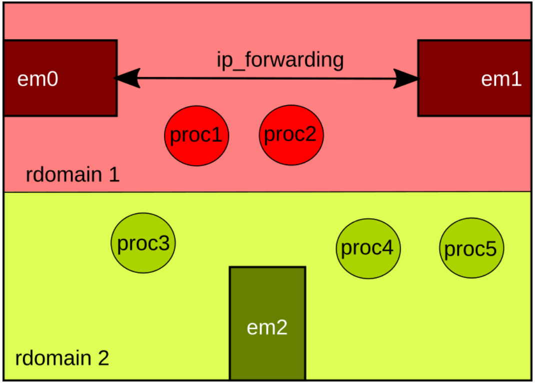
- the egress (WAN) interface can’t reach the internet
  - it may not even have an IP address
  - it cannot have a default route through that interface (conflict with the Carp route if it becomes active)
- there may be similar issues with the ingress (LAN) interface
- even if one adds a dedicated management interface, egress is still an issue

OpenBSD solution to this:
→ use routing domains to separate admin and packet filtering / routing.

Routing domains

Several independant routing tables in the system

- Each processus is bound to a **routing domain** with several routing tables.
- Network interfaces are bound to a routing domain
- Allows to completely separate network traffics



Important routing domain commands

- in `/etc/hostname.if` : `rdomain n` puts the interface in the given routing domain
- `id -R` displays the routing domain of the current shell
- `route -Tn exec command...` executes the given command in the specified routing domain
- `ps aux -o rtable` shows the routing table used by each process
- `rcctl set daemon rtable n` tells `rc.d` to start the given daemon in the specified routing table
- PF can be used to route packets betwen domains.

Routing domains for a Carp firewall

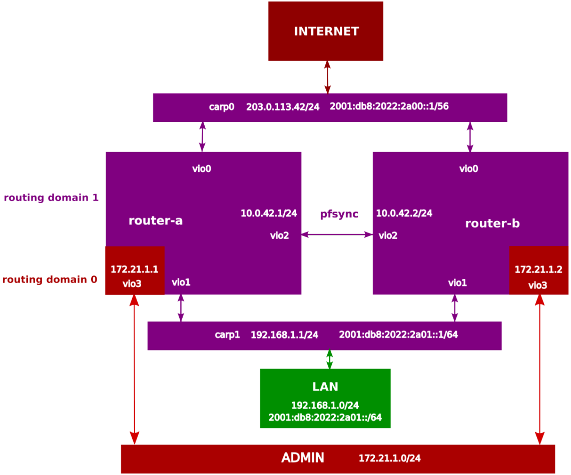
Copying the design of “real” routers using 2 routing domains:

Control: for admin, including a default route
Data: for packets filtering/forwarding/routing

From the sysadmin point of vue: use routing domain 0 as the control plane:

- doesn’t know about Carp
- has all the regular services (sshd, ntpd, unwind, ...) running
- has its own IP addresses and default route.

Carp with routing domains architecture



Installation and configuration

- connect the administration interface first
- install OpenBSD normally, configuring only this interface
- once firewalls are installed, the data routing domain can be configured

This procedure also makes it easy to prepare new firewalls that are going to replace existging ones.

/etc/sysctl.conf

On both routers, enable:

- packet forwarding (both v4 and v6)
- Carp preemption

/etc/sysctl.conf

```
net.inet.ip.forwarding=1
net.inet6.ip6.forwarding=1
net.inet.carp.preempt=1
```

Physical interfaces

/etc/hostname.vio0 et /etc/hostname.vio1 on both routers

```
up
rdomain 1
```

routeur A : /etc/hostname.vio2

```
rdomain 1
up
inet 10.0.42.1 255.255.255.0
```

routeur B : /etc/hostname.vio2

```
rdomain 1
up
inet 10.0.42.2 255.255.255.0
```

(pfsync transport is v4 only)

CARP: router A

/etc/hostname.carp0

```
rdomain 1
vhid 10 carpdev vio0 pass SuperSecret10
inet 203.0.113.42 255.255.255.255
inet6 2001:db8:2022:2a00::1/56
! route -T1 add default 203.0.113.1
! route -T1 add -inet6 default fe80::1%carp0
```

/etc/hostname.carp1

```
rdomain 1
vhid 20 carpdev vio1 pass SuperSecret20
inet 192.168.1.1 255.255.255.0
inet6 2001:db8:2022:2a01::1/64
```

CARP: router B

/etc/hostname.carp0

```
rdomain 1
vhid 10 advskew 200 carpdev vio0 pass SuperSecret10
inet 203.0.113.42 255.255.255.255
inet6 2001:db8:2022:2a00::1/56
! route -T1 add default 203.0.113.1
! route -T1 add -inet6 default fe80::1%carp0
```

/etc/hostname.carp1

```
rdomain 1
vhid 20 advskew 200 carpdev vio1 pass SuperSecret20
inet 192.168.1.1 255.255.255.0
inet6 2001:db8:2022:2a01::1/64
```

pfsync

`/etc/hostname.pfsync0` on both routers

```
rdomain 1
up syncdev vio2
```

/etc/pf.conf

On both routers :

```
ext=carp0
ext_if=vio0
int=carp1
int_if=vio1
sync=pfsync0
sync_if=vio2
adm=vio3

set skip on { lo $int_if $adm }

set loginterface $ext_if
```

```
block log

# let pfsync go through
pass quick on $sync_if proto pfsync \
    keep state (no-sync)
pass quick on $sync_if proto icmp

# idem for carp
pass quick on $ext_if proto carp \
    keep state (no-sync)

# outbound traffic
pass out on $ext_if inet nat-to ($ext)
```

Running services in the data plane

Since the data plane is in rdomain 1, some services must run there too.
For example for a SOHO like configuration: dhcpd, rad, unbound, ntpd,...

```
dhcpd_flags=-y 10.0.42.1 -Y 10.0.42.2 carp1
dhcpd_rtable=1
rad_flags=
rad_rtable=1
unbound_flags=
unbound_rtable=1
ntpd_flags=
ntpd_rtable=1
```

LAAS setup

External firewalls pair:

- 2 Dell R340 with Xeon E-2124 (3.30 GHz) CPUs and 16GiB of RAM
- Dual Intel X710 10Gb/s cards for packet forwarding (rdomain 1)
- Dual Intel X550T cards for pfsync (rdomain 1) and management (rdomain 0)
- ca 320 PF rules

Internal firewalls pair:

- 2 Virtual machines running on a VMWare cluster with 1vCPU and 4GiB RAM
- using 2 vmxnet3 interfaces:
  - one in rdomain 0 for management
  - one in rdomain 1 carrying all the internals VLANs and Carp interfaces (26)
- ca 300 PF rules

All four systems running OpenBSD 7.3-stable currently.

PF rulesets

- Block everything inbound
- Allow specific trafic to specific servers (Web, SSH, SMTP/IMAP...)
- Let most of egress trafic out
- Use `max-src-conn` and `max-sec-conn-rate` to slow down brute-force attacks
- Use dynamic block lists (`https://feodotracker.abuse.ch/`) to block Botnets trafic
- Use pflog for mandatory logging of some outbound trafic

Updating pf rules

/usr/local/bin/update-pf

```
#! /bin/ksh
autre="routeur-b"
if /sbin/pfctl -n -f /etc/pf.conf; then
       echo "syntaxe ok -> update"
       pfctl -f /etc/pf.conf
else
       echo "erreur dans pf.conf"
       exit 2
fi
echo "mise a jour $autre"
scp -p /etc/pf.conf $autre:/etc/
ssh autre /sbin/pfctl -f /etc/pf.conf
exit 0
```

IPv6

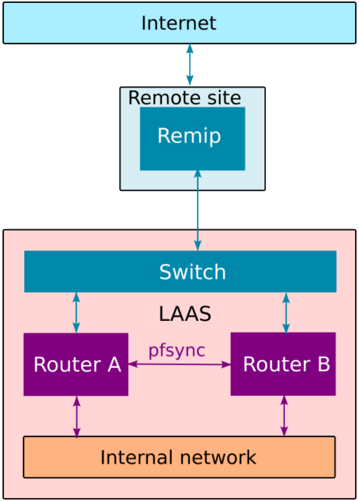
- makes rule sets bigger
- need special rules for Link-Local and Multicast trafic
- Carp over IPv6 only?
- Pfsync currently v4 only
- need to run *rad(8)* on the active router (→ *ifstated(8)* rules)

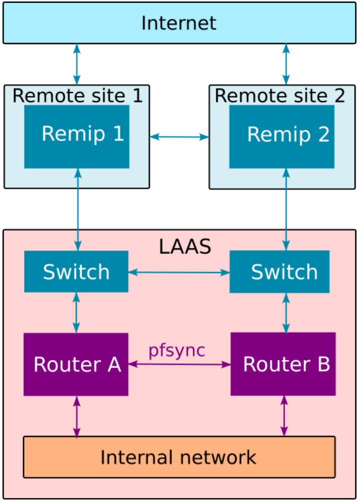
On Carp usage

- Initially : protect from kernel panics / hangs
- → very few (none) of such events
- also very few / no hardware failures
- main use : seemless syspatch / sysupgrades

There have been one strange failure where the facing uplink provider router would stop answering to ARP → loss of IPv4 connectivity, not detectable by Carp or ifstated

New external connection setup: single to double attachment



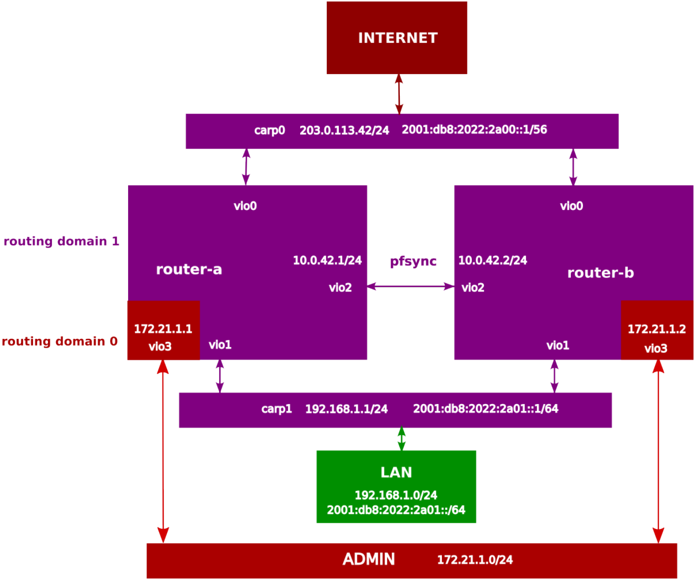


A test setup with *vmm(4)*

On a machine with 2 physical interfaces (or 1 interface and a vlan-capable switch):

- set up 2 virtual machines and 3 *veb(4)*
- bind the external interface to the internet
- bind the internal interface to the lan
- use the "local" vmm network for the admin rdomain

Makes it possible to test things on a laptop or a smallish machine before going to production.


INTERNET
carp0 203.0.113.42/24 2001:db8:2022:2a00::1/56
vio0
vio0
routing domain 1
router-a
10.0.42.1/24
pfsync
10.0.42.2/24
router-b
vio2
vio2
172.21.1.1
vio3
routing domain 0
vio1
vio1
172.21.1.2
vio3
carp1 192.168.1.1/24 2001:db8:2022:2a01::1/64
LAN
192.168.1.0/24
2001:db8:2022:2a01::/64
ADMIN 172.21.1.0/24

VMM network setup

/etc/vm.conf (1/3)

```
switch ext {
      enable
      interface veb0
}

switch int {
      enable
      interface veb1
}

switch pfsync {
      enable
      interface veb2
}
```

router-a VM config - /etc/vm.conf (2/3)

```
vm "router-a" {
   disable
   memory 1G
   disk "/local/vm/a.img"
   interface {
      switch "ext"
   }
   interface {
      switch "int"
   }
   interface {
      switch "pfsync"
   }
   local interface
   owner matthieu
}
```

router-b VM config - /etc/vm.conf (3/3)

```
vm "router-b" {
    disable
    memory 1G
    disk "/local/vm/b.img"
    interface {
        switch "ext"
    }
    interface {
       switch "int"
    }
    interface {
        switch "pfsync"
    }
    local interface
    owner matthieu
}
```

Conclusion

- Running OpenBSD firewalls at LAAS for almost 15 years
- On a 10Gb/s uplink since 3 years.
- Performance is not a concern (good enough)
- The use of routing domains to separate admin interface is really comfortable
- Have successfully blocked a few mild DDoS attacks
- PF states and debugging tools provide good insight into actual network flows
- Main concern: teach staff about OpenBSD and pf administration
- Also: hardware compatibility issues when replacing the hardware

Questions ?