

OpenBSD and Linux: Insights into a migration project at the INI

Stephan A. Rickauer, LinuxTag '07

Introduction

The IT Infrastructure at INI

Problems with our Linux Setups

The Migration to OpenBSD

The Outcome / Summary

Post Migrational Issues

Future Plans

Conclusion

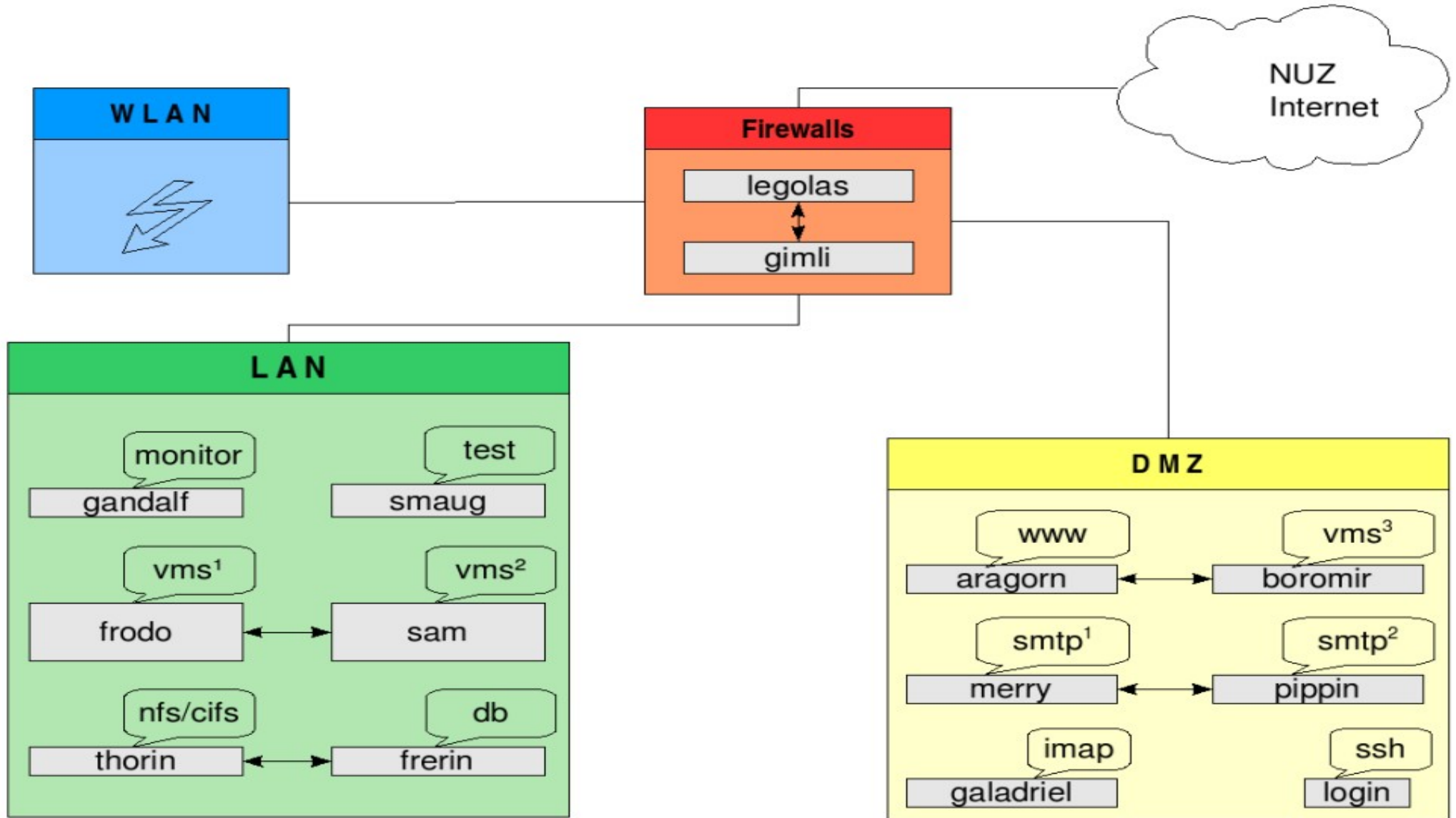
- The Institute of Neuroinformatics
 - Established in 1995
 - At University and ETH Zurich
 - ~ 80 researchers / 3 professors

“The mission of the Institute is to discover the key principles by which brains work and to implement these in artificial systems that interact intelligently with the real world.”



- Heterogeneous IT environment
 - 120 Linux workstations
 - 20 Windows PC's
 - 20 Servers
 - 30 Special purpose machines
 - 1.45 work equivalents
- Four network segments
 - LAN, WLAN, DMZ, NUZ (Uplink)

Network overview



- Main HA technologies used ...
 - Heartbeat
 - DRBD
 - Netfilter (iptables)
- ... on active/passive clusters
 - File servers (frodo/sam, SuSE 9.0)
 - Mail servers (merry/pippin, SuSE 9.1)
 - Firewalls (legolas/gimli, SuSE 9.0)

Problems with our Linux setups: technical issues

- Technical issues

- DRBD

- Very robust, but OS upgrades are critical
 - Patching on the 'active' node only
 - Long sync times prior to 9.1 (fixed with 'QuickSync')

- heartbeat

- Slow: takeover up to 20s due to ARP broadcasts
 - Lost 'sessions', reconnects required

- Netfilter

- No session synchronisation

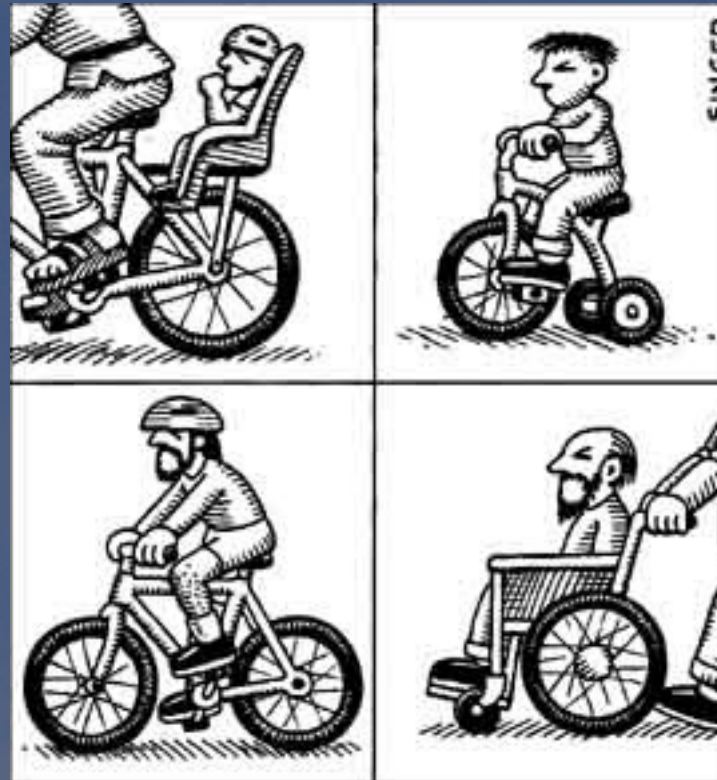
Problems with our Linux setups: lifecycle

- Lifecycle issues

- Community releases vs. 'business releases'
- The price of a long(er) lifecycle

“Software of quality focused projects becomes more secure over time.”

*Andy Ozment, Stuart E. Schlechter
“Milk or Wine: Does Software Security Improve with Age?”, MIT Lincoln Lab.*



Why the hell OpenBSD?



Why OpenBSD: Concepts

Secure by default



Knobs suck

Enforce open standards

Free, as in blob-free

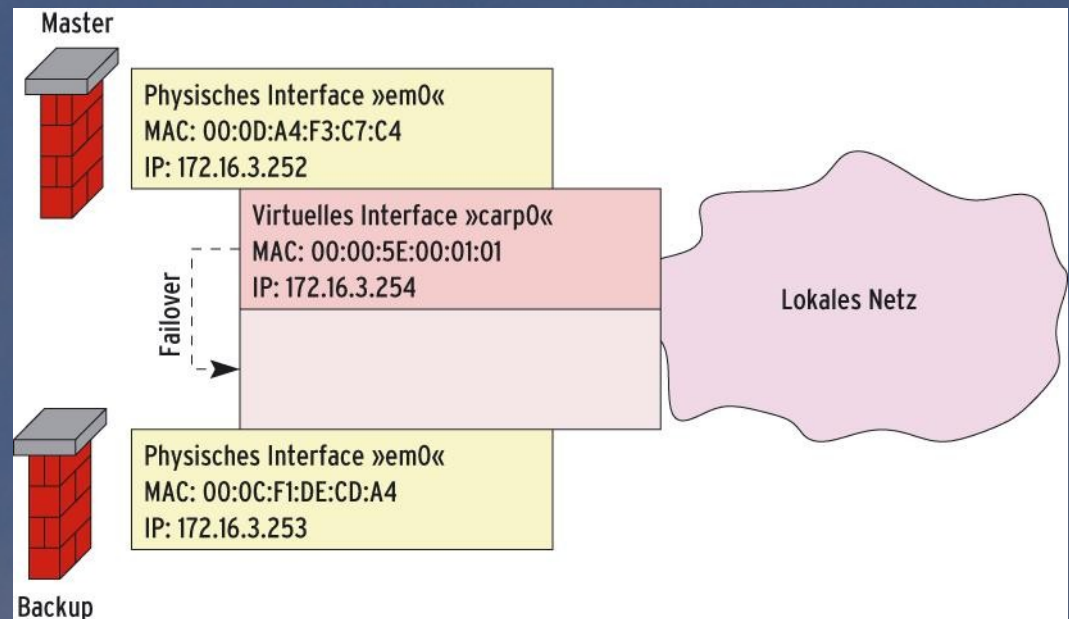
The migration to OpenBSD: Lifecycle

- Lifecycle
 - Short (one year):
 - Fast and steady evolution
 - Quality driven improvements
 - Fixed (1st of May / Nov.)
 - Very easy to plan (budget, time, support)
 - Smart
 - Upgrade procedures very well documented
 - Upgrades possible with no physical access
 - 30min down to 10min per machine

The migration to OpenBSD: CARP

● CARP

- Common Address Redundancy Protocol
- Free implementation of Cisco's HSRP/VRRP
- Covers layer 2 *and* 3
- Quick!
- unlimited nodes
- active/passive
- active/active
(ARP loadbalancing)



The migration to OpenBSD: pf/pfsync

- pf (Packet Filter)

- Human readable syntax

- “pass in inet proto tcp from any to \$if port ssh”

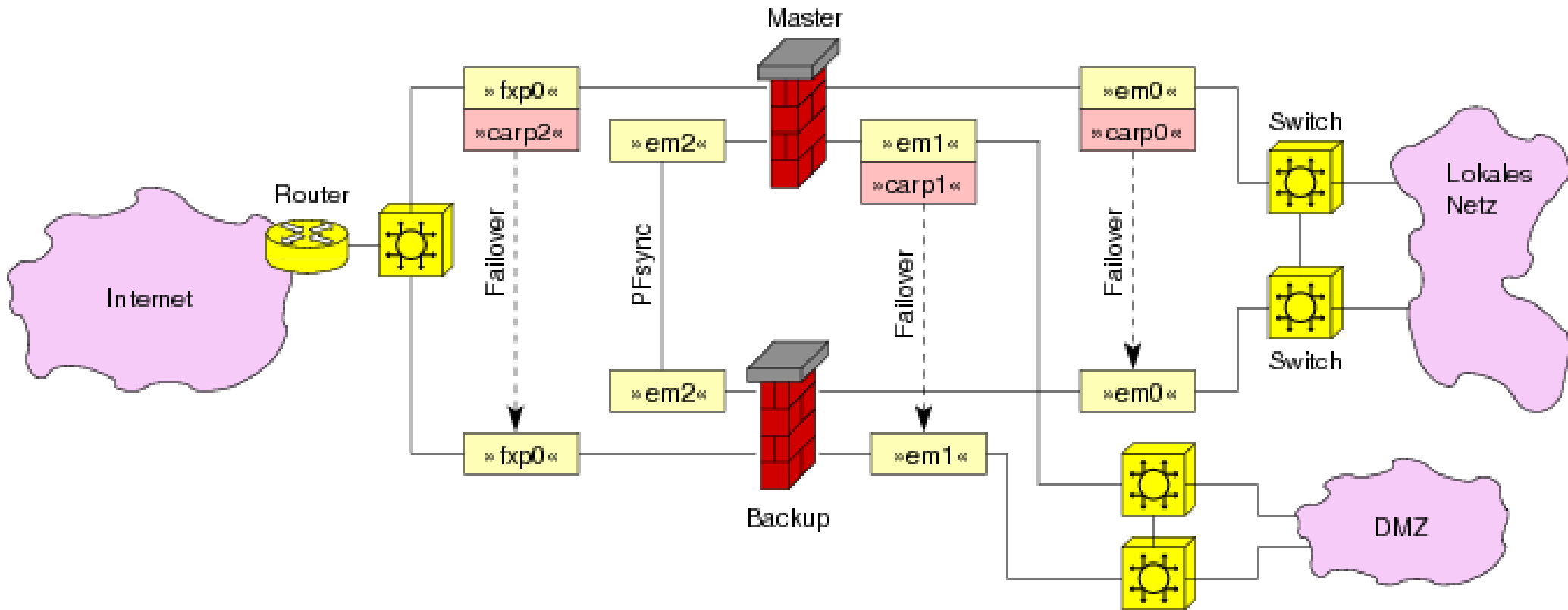
- Advanced feature set

- Logging via pflogd

- pfsync

- Session synchronisation

The migration to OpenBSD: pf/pfsync @INI

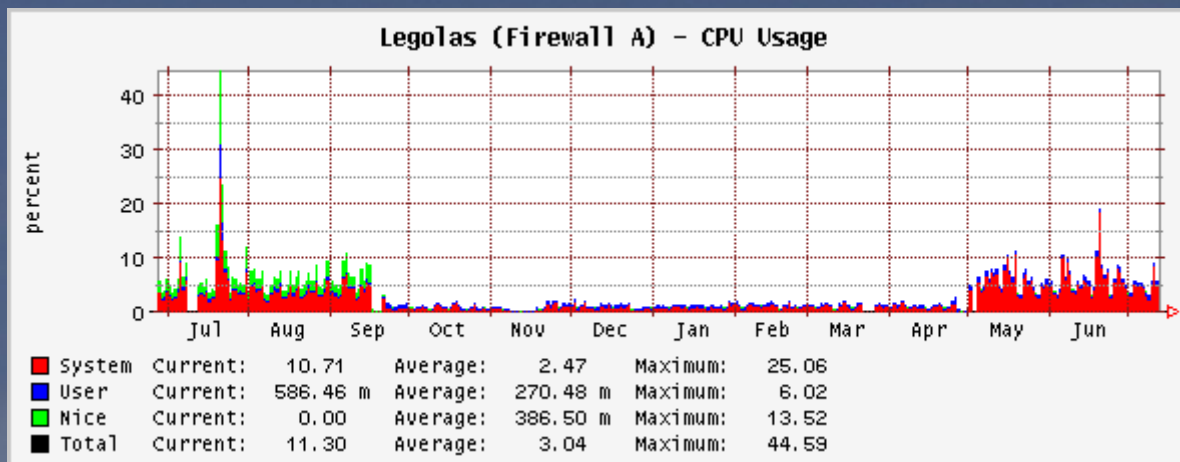


- Unforeseen problems
 - ftp-proxy in 3.7
 - Flavours: -stable vs. -current
- Why we can't use OpenBSD everywhere
 - iSCSI
 - TSM Backup (Tivoli)
 - LVM / file system

The outcome

Legolas	Firewall	SuSE 9.0	OpenBSD 4.0-stable
Gimli	Firewall	SuSE 9.0	OpenBSD 4.0-stable
Merry	SMTP/ext. DNS	SuSE 9.1	OpenBSD 4.0-stable*
Pippin	IMAPs	SuSE 9.1	SLES 10
Frodo	NFS/Samba	SuSE 9.0	SLES 10
Sam	NFS/Samba	SuSE 9.0	SLES 10
Balin	Int. DNS/DHCP	SuSE 9.0	OpenBSD 4.0-stable
Dwalin	Int. DNS/DHCP	SuSE 9.0	OpenBSD 4.0-stable
Bombadil	CUPS	SuSE 9.0	OpenBSD 4.0-current
Arwen	Subversion	FreeBSD 4.x	OpenBSD 4.0-current
Kobur	RPM/apt-get server	n/a	OpenBSD 3.9-release
Elrond	Build host	n/a	OpenBSD 4.0-stable
Soekris	SSH login	FreeBSD 4.x	OpenBSD 4.0-stable

- Things we realised after the migration
 - Great community support
 - max-src-conn-rate
 - release(8)
 - High quality documentation
 - Efficient use of system resources (virtualisation!)



Linux “netstat”

```
xterm enomaly@gamealmighty:~ xterm<1> Virtual Users And Domains With P... bash|enomaly@gamealmighty/op...
NETSTAT(8) Linux Programmer's Manual NETSTAT(8)

NAME
netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

SYNOPSIS
netstat [address_family_options] [--tcp|-t] [--udp|-u] [--raw|-r] [--listening|-l] [--all|-a] [--numeric|-n]
[--numeric-hosts] [--numeric-ports] [--numeric-ports] [--symbolic|-N] [--extend|-e] [--extend|-e] [--timers|-o]
[--program|-p] [--verbose|-v] [--continuous|-c] [delay]

netstat [--route|-r] [address_family_options] [--extend|-e] [--extend|-e] [--verbose|-v] [--numeric|-n] [--numeric-
hosts] [--numeric-ports] [--numeric-ports] [--continuous|-c] [delay]

netstat [--interfaces|-i] [iface] [--all|-a] [--extend|-e] [--extend|-e] [--verbose|-v] [--program|-p]
[--numeric|-n] [--numeric-hosts] [--numeric-ports] [--numeric-ports] [--continuous|-c] [delay]

netstat [--groups|-g] [--numeric|-n] [--numeric-hosts] [--numeric-ports] [--numeric-ports] [--continuous|-c] [delay]

netstat [--masquerade|-M] [--extend|-e] [--numeric|-n] [--numeric-hosts] [--numeric-ports] [--numeric-ports] [--con-
tinuous|-c] [delay]

netstat [--statistics|-s] [--tcp|-t] [--udp|-u] [--raw|-r] [delay]

netstat [--version|-V]

netstat [--help|-h]

address_family_options:
[--protocol={inet,unix,ipx,ax25,netrom,ddp}[...]] [--unix|-x] [--inet|-ip] [--ax25] [--ipx] [--netrom] [--ddp]

DESCRIPTION
Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by
the first argument, as follows:

(none)
By default, netstat displays a list of open sockets. If you don't specify any address families, then the active
sockets of all configured address families will be printed.

--route, -r
Display the kernel routing tables.

--groups, -g
Display multicast group membership information for IPv4 and IPv6.

--interface=iface, -i
Display a table of all network interfaces, or the specified iface.

--masquerade, -M
Display a list of masqueraded connections.

[ Fri 2006-09-15 00:11 | load: 0.26, 0.18, 0.20 ]
```

OpenBSD “netstat”

```
xterm enomaly@gamealmighty:~ xterm<1> xterm<2> Virtual Users And Domains ... bash|enomaly@gamealm...
NETSTAT(1) OpenBSD Reference Manual NETSTAT(1)

NAME
netstat - show network status

SYNOPSIS
netstat [-Ran] [-f address_family] [-M core] [-N system]
netstat [-bdgilnqrstu] [-f address_family] [-M core] [-N system]
netstat [-bdn] [-I interface] [-M core] [-N system] [-u wait]
netstat [-M core] [-N system] -P pcbaddr
netstat [-s] [-M core] [-N system] [-p protocol]
netstat [-a] [-f address_family] [-i i] [-I interface]
netstat [-M interface]

DESCRIPTION
The netstat command symbolically displays the contents of various net-
work-related data structures. There are a number of output formats, de-
pending on the options for the information presented.

The first form of the command displays a list of active sockets for each
protocol. The second form presents the contents of one of the other net-
work data structures according to the option selected. Using the third
form, with a wait interval specified, netstat will continuously display
the information regarding packet traffic on the configured network inter-
faces. The fourth form displays statistics about the protocol control
block (PCB). The fifth form displays statistics about the named proto-
col. The sixth form displays per interface statistics for the specified
address family. The final form displays per interface statistics for the
specified wireless (802.11) device.

The options are as follows:

-R
With the default display, show the address of any protocol con-
trol blocks associated with sockets; used for debugging, e.g.
with the -P flag.

-a
With the default display, show the state of all sockets; normally
sockets used by server processes are not shown. With the inter-
face display (options -I or -i), show multicast addresses.

-b
With the interface display (options -I or -i), show bytes in and
out, instead of packet statistics.

-d
With either the interface display (options -I or -i) or an inter-
val (option -u), show the number of dropped packets.

-f address_family
Limit statistics or address control block reports to those of the
specified address_family.

The following address families are recognized:
/usr/share/man/cat1/netstat.0.23%

[ Fri 2006-09-15 00:11 | load: 0.30, 0.20, 0.20 ]
```

- hoststated
 - Monitor hosts
 - keep related pf rules up to date
- ifstated
 - Monitor network state changes
 - Change running services

- Original plan to migrate only two firewalls has led to eight machines being migrated to OpenBSD.
- New OpenBSD machines have been introduced and there are likely to be more to come.
- Maintenance overhead was drastically reduced.
- Overall security and availability was increased.

OpenBSD has become
our strategic platform no. 1!

Thank you for your attention.