BSD Overview

Jim Brown
May 24, 2012

BSD Overview

I – A Brief History of BSD

- ATT UCB Partnership
- ATT(USL) Lawsuit
- BSD Family Tree
- BSD License

II – The Core BSD Projects

- Main Features
- Community
- Future Directions

III – Cool Hot Stuff

- Batteries Included
- ZFS , Hammer
- pf Firewall, pfSense
- Capsicum
- Virtualization Topics
  - Jails, Xen, etc.
- Desktop PC-BSD

IV – BSD Certification

A *(Very)* Brief History of BSD

- **1971** – ATT cheaply licenses Unix source code to many organizations, including UCB as educational material

- **1975** – Ken Thompson takes a sabbatical from ATT, brings the latest Unix source on tape to UCB his alma mater to run on a PDP 11 which UCB provided. (Industry/academic partnerships were much more common back then.)

- Computer Science students (notably Bill Joy and Chuck Haley) at UCB begin to make numerous improvements to Unix and make them available on tape as the "Berkeley Software Distribution" - BSD

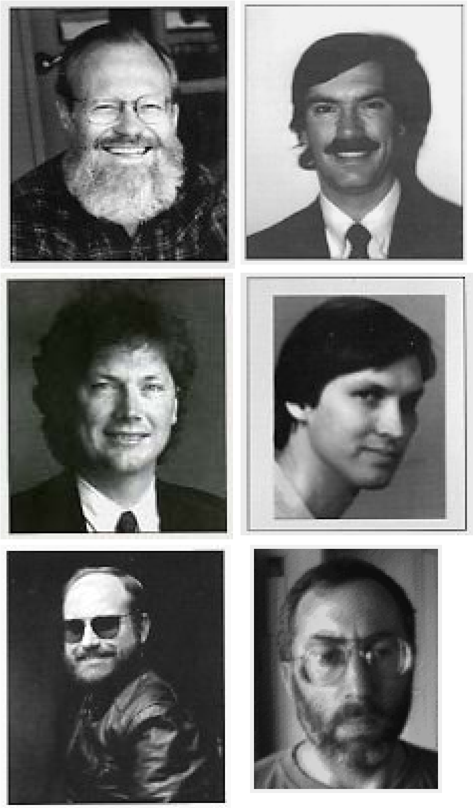
A *(Very)* Brief History of BSD

- **1980** – Computer Science Research Group (CSRG) forms at UCB with DARPA funding to make many more improvements to Unix - job control, autoreboot, fast filesystem, gigabit address space, Lisp, IPC, sockets, TCP/IP stack + applications, r* utils, machine independence, rewriting almost all ATT code with UCB/CSRG code, including many ports

- **1991** – The Networking Release 2 tape is released on the Internet via anon FTP. A 386 port quickly follows by Bill and Lynne Jolitz. The NetBSD group is formed- the first Open Source community entirely on the Internet

- **1992** – A commercial version, BSDI (sold for $995, 1-800-ITS-UNIX) draws the ire of USL/ATT. USL sues BSDI and UCB. The lawsuit was resolved in Jan, 1994.

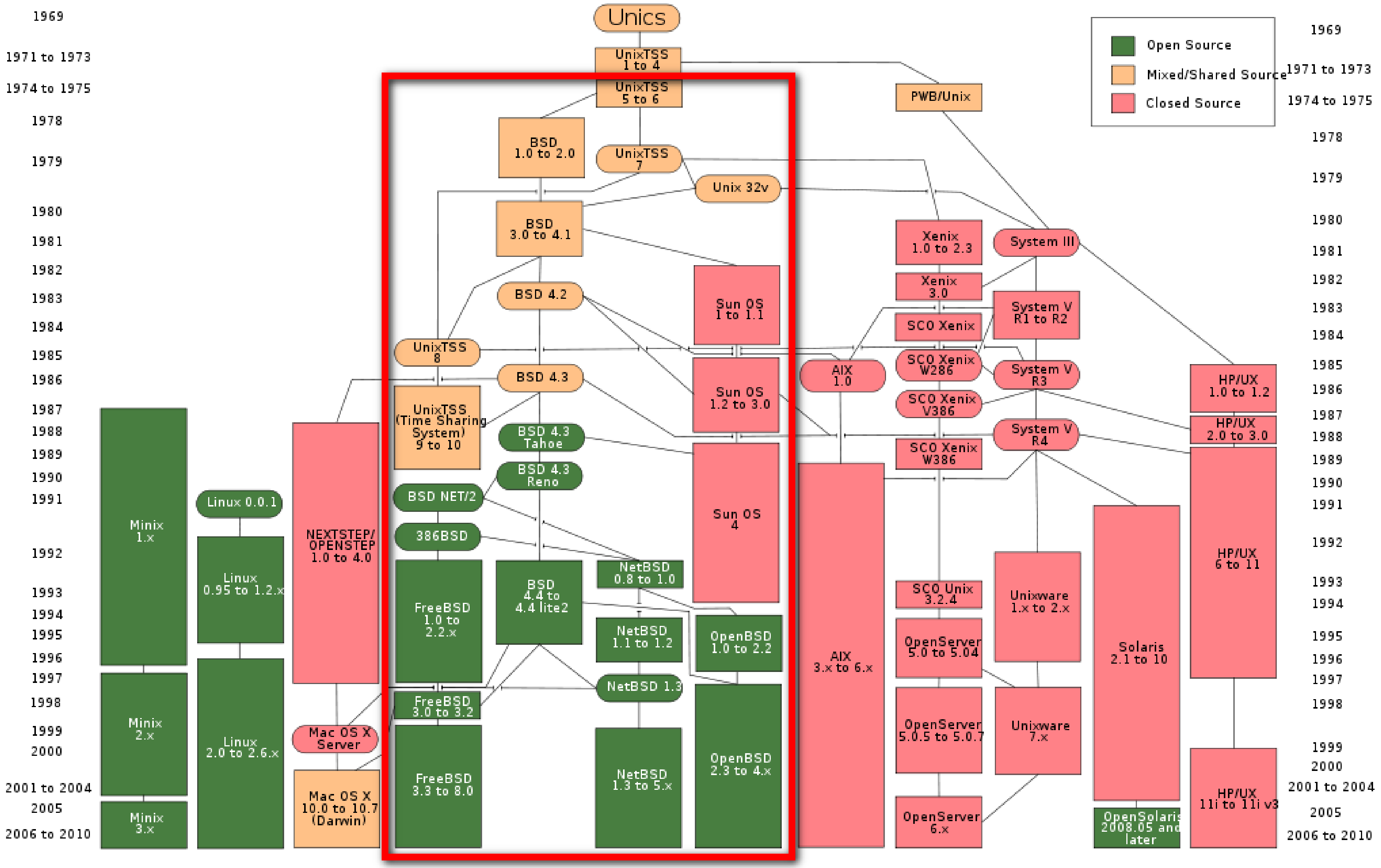
Some notable CSRG members



Fabry
Joy
Bostic

McKusick
Karels
Leffler

Unix Family Tree



BSD License (3 clause)

Copyright (c) <year>, <copyright holder>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

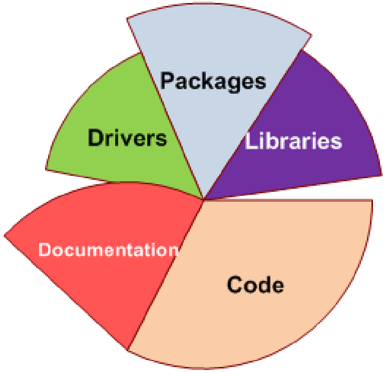
Don’t pretend you wrote it.

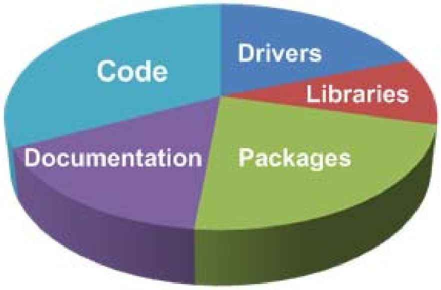
Software is provided “AS IS” – i.e. don’t sue the developers if you don’t like the way it works.

BSD Projects – No Assembly Required

Linux / BSD differences -

- Linux is a kernel, with additional packaging by people of sometimes varying capabilities. As a result, Linux often feels mismatched – like pieces stuck together – with important pieces sometimes altogether missing, or not well integrated.
- This is less true today with corporate backing, but the refined versions cost $$$.





- With BSD projects, you get the entire collection, fully integrated, for free.
- Corporate sponsorship happens too, mostly internals and documentation. The projects are responsible for the released product.
- BSDs sometimes lack commercial driver support. Controversy over binary blob drivers in the BSD community remains an issue.

Core BSD Projects – FreeBSD

- **FreeBSD** – most widely known, solid academic roots. Widely used in commercial systems, ISPs, large scale systems. Research platform for many ideas and protocols
- Focus on production ready systems ("The Power to Serve")

- Main features (standard): Dtrace, Large Scale SMP support, SMP aware TCP/IP, modular TCP congestion algorithms, SIFTR, Ipv6 only kernel available, CLANG/LLVM compiler, linuxulator (See also Cool Hot Stuff)

- Community – > 1000 developers, ~300 committers, ~10 core team (9 elected members and 1 selected secretary); ~100 main projects, ~ 50-100 commits per day; Multiple cvs/svn branches

- Future directions – Virtualization, embedded, enhanced networking support (NetMap)

- www.freebsd.org , www.freebsdfoundation.org - a 501(c)(3) organization

Core BSD Projects - NetBSD

- **NetBSD** – Oldest open source BSD project , solid academic roots, widely used in research systems
- Focus on portability (“Of course it runs NetBSD”), 8 Tier I, 49 Tier II ports.
  - Cross platform build tool (build.sh)
  - Reference arch for Xen on BSD (domU and dom0)

- Main features (standard): OS emulation (11 variants), UVM system, noexec stack and heap, modular kernel authorization framework (kauth), verifiedexec

- Community – ~500 developers, ~150 committers, ~ 8 historical core team, 50+ ports , ~ 10-30 commits per day

- Future directions – Embedded systems, high speed networking, more ports, pkgsrc

- www.netbsd.org ,www.netbsd.org/foundation – a 501(c)(3) organization

Core BSD Projects – OpenBSD

- **OpenBSD** – Fork of NetBSD in 1995 with focus on portability, standardization, correctness, proactive security and integrated crypto.
- Focus on secure code (“Only two remote holes in the default install, in a heck of a long time!” and “Secure by default.”)
  - First use of anonymous CVS in the OSS community
  - Open, transparent code (no blobs, no NDAs)
  - Minimal (secure) installation
  - ‘Spartan’ installer, often ridiculed
- Main features (standard): continuous security and license audit, ProPolice, W^X, least privilege, ASLR, pf, (see Cool Hot Stuff)

- Community – a few hundred developers, ~130 committers, 1 project leader (Theo de Raadt); a handful of main projects, ~10-30 commits per day; Releases every six months (like clockwork)
- OpenBSD is based out of de Raadt’s home in Canada

- Future directions – Secure code and releases, integrated cryptography. Significant associated projects: OpenSSH, OpenBGP, OpenNTP, and OpenCVS.

- www.openbsd.org , www.openbsdfoundation.org – a Canadian non-profit

Core BSD Projects – DragonFly BSD

- **DragonFly BSD** – Fork of FreeBSD 4.8 in 2003, by Matt Dillon. Research platform for many ideas; logical continuation of FreeBSD 4.8
- Focus originally on clustering, and sophisticated cache management
- Main features – LWKT (Light Weight Kernel Thread scheduler), DEVFS, VKERNEL ,NULLFS - NULL Filesystem Layer, Process Checkpointing, SWAPCACHE - Managed SSD support, Transparent disk encryption
- Community – ~75 developers, ~50 committers, 1 project leader – Matt Dillon; ~ 5-20 commits per day
- Future directions – Virtualization, embedded, enhanced networking support (NetMap)
- www.dragonflybsd.org (foundation status unknown)

Cool

HOT

Stuff

BSD Cool Hot Stuff

- **ZFS** - unique combination of filesystem and volume manager for disk storage

  – Designed to combat silent corruption with extensive checksumming
  – Large maximum size making it ideal for Big Data.
      Max size is 2^64 bytes (16 exabytes)
  – Pools via virtual devices, can be grown dynamically and combined with RAID
  – Transparent compression, encryption (currently Oracle only), deduplication
  – ZFS cache - very high performance, uses RAM, and disk in optimal fashion.
      Separate read, write caches.
  – Snapshots and writable snapshots (clones)
  – Deduplication can reduce the overall storage requirements for ZFS
  – Dynamic striping – as new devices are added to the zpool,
      they are included in the striping
  – Dynamic repair (scrub and resilvering)
  – ZFS has been available since FreeBSD 7.0, 2007. Since FreeBSD 8.3, zpool
    version is latest version (28).
  – License issues are hampering ZFS on Linux although a FUSE
      product is in development (zfs-fuse.net)

BSD Cool Hot Stuff

- **HAMMER** – DragonFly BSD filesystem developed by Matt Dillon
  - Released with DragonFly BSD 2.0 in 2008.

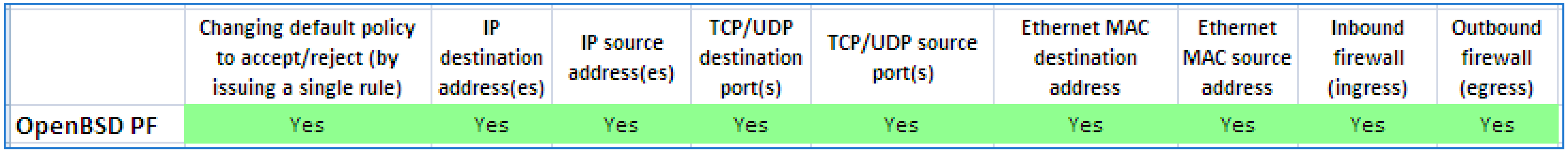
  Size
  - HAMMER file system can be up to 1 exabyte in size, and can encompass up to 256 volumes, each of which can be up to 4 PB

  Snapshots
  - HAMMER retains a fine-grained history, and coarse-grained history (snapshots)
  - Are live, and can be taken at any time
  - May be used to access entire directory trees
  - Indexed by the on-media B-Tree and are extremely storage-efficient.

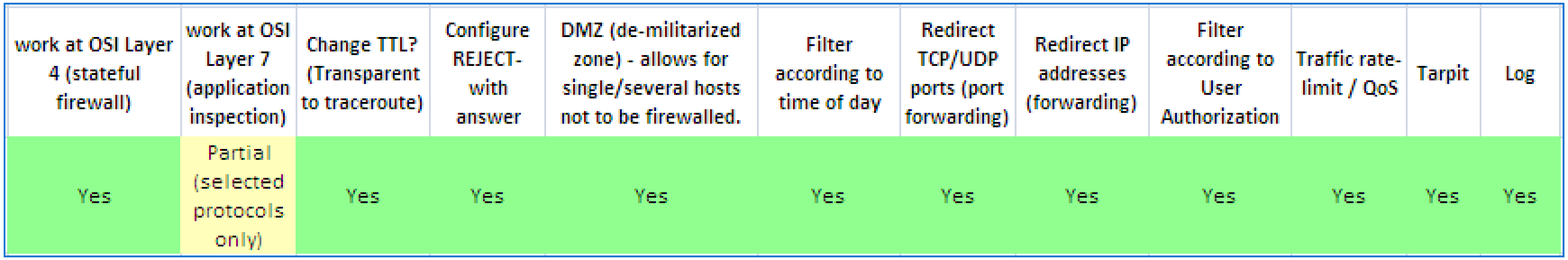
  Backups and history
  - HAMMER file systems can be split up into multiple pseudo-file systems, or PFSs. Snapshots and backups can be different for each individual PFS.
  - HAMMER PFSs can be backed up continuously or in batch to other HAMMER PFSs, on a per-PFS basis.
  - Backup PFSs (slaves) are functionally identical to the original (master) and can be promoted to a master.
  - Slave PFSs can retain file history independent of the master volume's settings.
  - HAMMER can efficiently stream bandwidth-controlled near-real-time backup data to slave PFSs on remote hosts.

BSD Cool Hot Stuff

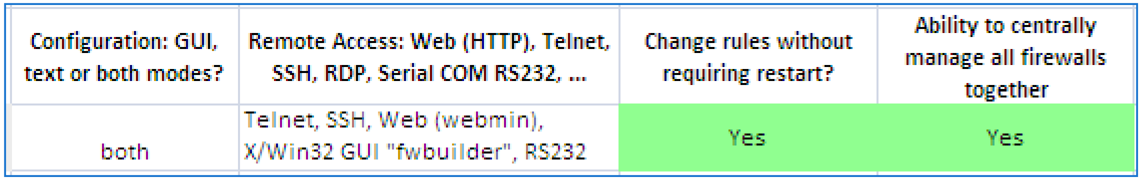
- OpenBSD **pf** Firewall – developed by Daniel Hartmeier and a host of others
  - Released in 2001 with OpenBSD 3.0. Also runs on Free, Net, and Dfly.
  - Mature firewall with extensive feature set and intuitive syntax.



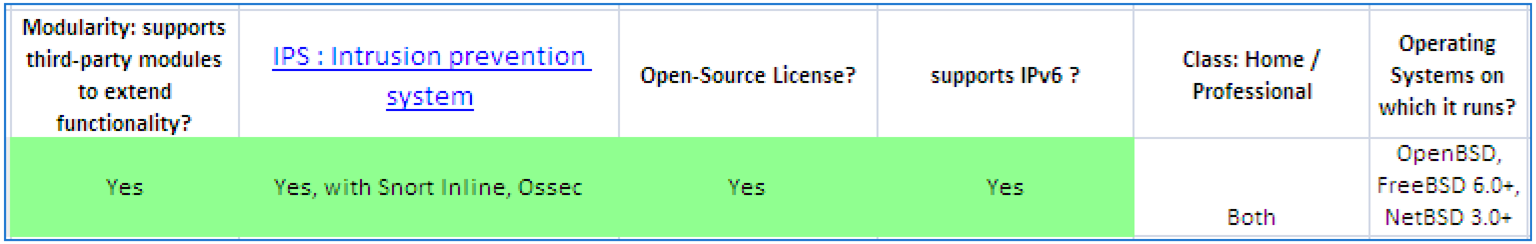
	Changing default policy to accept/reject (by issuing a single rule)	IP destination address(es)	IP source address(es)	TCP/UDP destination port(s)	TCP/UDP source port(s)	Ethernet MAC destination address	Ethernet MAC source address	Inbound firewall (ingress)	Outbound firewall (egress)
OpenBSD PF	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes



work at OSI Layer 4 (stateful firewall)	work at OSI Layer 7 (application inspection)	Change TTL? (Transparent to traceroute)	Configure REJECT-with answer	DMZ (de-militarized zone) - allows for single/several hosts not to be firewalled.	Filter according to time of day	Redirect TCP/UDP ports (port forwarding)	Redirect IP addresses (forwarding)	Filter according to User Authorization	Traffic rate-limit / QoS	Tarpit	Log
Yes	Partial (selected protocols only)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes



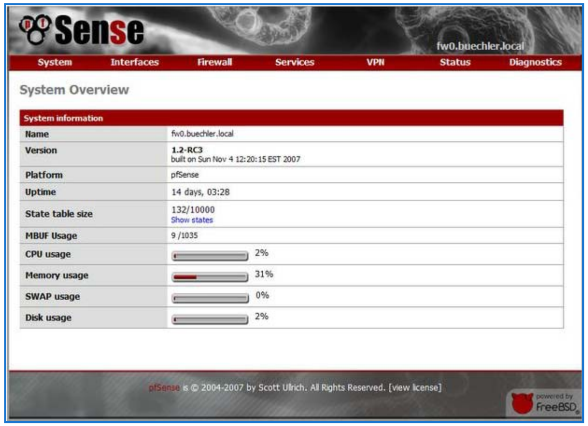
Configuration: GUI, text or both modes?	Remote Access: Web (HTTP), Telnet, SSH, RDP, Serial COM RS232, ...	Change rules without requiring restart?	Ability to centrally manage all firewalls together
both	Telnet, SSH, Web (webmin), X/Win32 GUI "fwbuilder", RS232	Yes	Yes



Modularity: supports third-party modules to extend functionality?	[IPS : Intrusion prevention system]()	Open-Source License?	supports IPv6 ?	Class: Home / Professional	Operating Systems on which it runs?
Yes	Yes, with Snort Inline, Ossec	Yes	Yes	Both	OpenBSD, FreeBSD 6.0+, NetBSD 3.0+

BSD Cool Hot Stuff

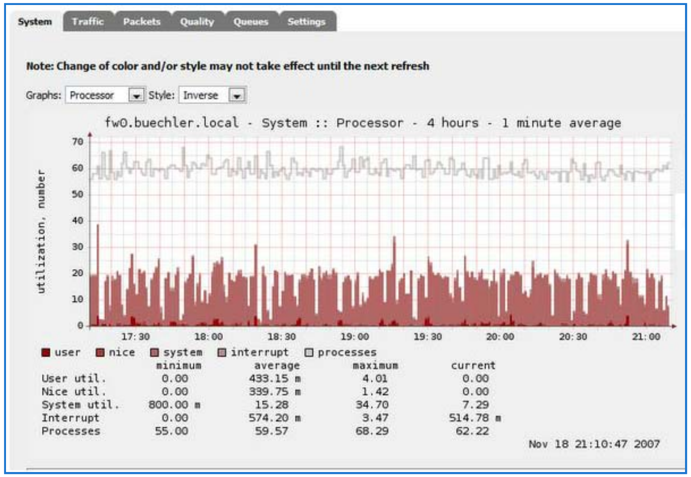
- **pfSense** - commercially supported firewall product based on pf. Fork of m0n0wall, released in 2004. Currently over 100,000 live installs. Full community support- forums, wiki, etc.



System Overview

System information	
Name	fw0.buechler.local
Version	1.2-RC3 built on Sun Nov 4 12:20:15 EST 2007
Platform	pfSense
Uptime	14 days, 03:28
State table size	132/10000 Show states
MBUF Usage	9 /1035
CPU usage	2%
Memory usage	31%
SWAP usage	0%
Disk usage	2%

pfSense is © 2004-2007 by Scott Ullrich. All Rights Reserved. [view license]

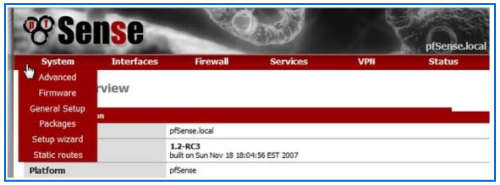


Note: Change of color and/or style may not take effect until the next refresh

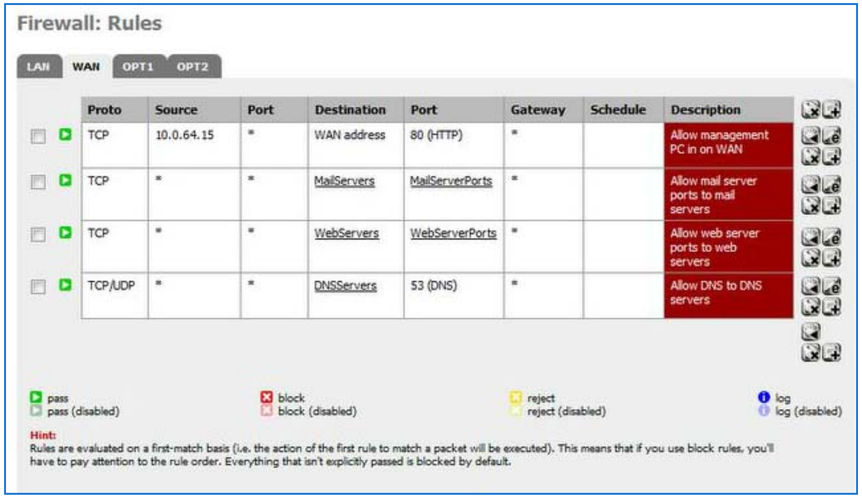
Graphs: Processor Style: Inverse

```
fw0.buechler.local - System :: Processor - 4 hours - 1 minute average

■ user  ■ nice  ■ system  ■ interrupt  □ processes
                   minimum       average       maximum       current
User util.          0.00      433.15 m          4.01          0.00
Nice util.          0.00      339.75 m          1.42          0.00
System util.      800.00 m     15.28           34.70          7.29
Interrupt           0.00      574.20 m          3.47        514.78 m
Processes          55.00       59.57           68.29         62.22
                                                        Nov 18 21:10:47 2007
```



	pfSense.local
	1.2-RC3 built on Sun Nov 18 18:04:56 EST 2007
Platform	pfSense



Firewall: Rules

		Proto	Source	Port	Destination	Port	Gateway	Schedule	Description
□	▶	TCP	10.0.64.15	*	WAN address	80 (HTTP)	*		Allow management PC in on WAN
□	▶	TCP	*	*	MailServers	MailServerPorts	*		Allow mail server ports to mail servers
□	▶	TCP	*	*	WebServers	WebServerPorts	*		Allow web server ports to web servers
□	▶	TCP/UDP	*	*	DNSServers	53 (DNS)	*		Allow DNS to DNS servers

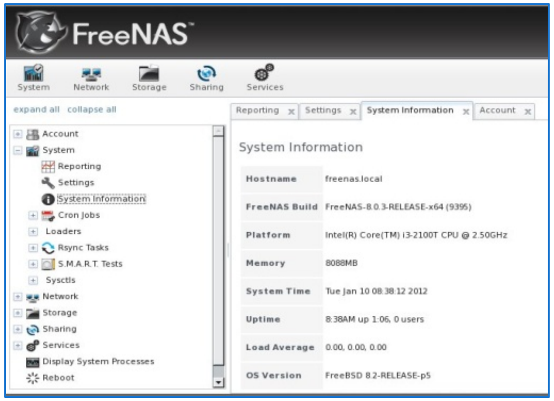
▶ pass ▷ pass (disabled) ☒ block ☒ block (disabled) □ reject □ reject (disabled) ● log ○ log (disabled)

Hint:
Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

BSD Cool Hot Stuff

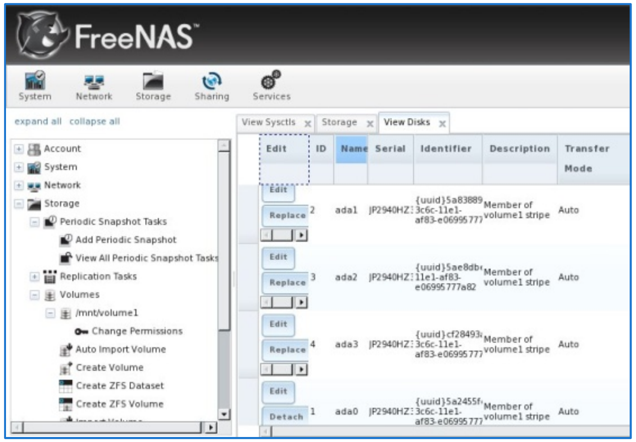
- **FreeNAS, TrueNAS –** Network Attached Storage software (FreeNAS) and appliances (TrueNAS). Free version and commercially supported version. Extensive features and wide community support.

•Supports AFP, CIFS, FTP, NFS, SSH (including SFTP), and TFTP as file sharing mechanisms
•Supports exporting file or device extents via iSCSI
•Supports Active Directory or LDAP for user authentication
•Supports UFS2 based volumes, including gmirror, gstripe, and graid3
•Supports ZFS as the primary filesystem, enabling many features not available in UFS2 such as quotas, snapshots, compression, replication, and datasets for sharing subsets of volumes
•Upgrade procedure takes advantage of NanoBSD by writing the operating system to an inactive partition, allowing for an easy reversal of an undesirable upgrade
•Django-driven graphical user interface
•rsync configuration through the graphical interface
•Menu localization
•UPS management in GUI
•USB 3.0 support
•ACLs and UNIX file system permissions work properly on both UFS and ZFS volumes
•Periodic ZFS snapshots are now exported to CIFS shares and are visible in Windows as shadow copies
•Netatalk (AFP) is now compatible with OS X 10.7



System Information

Hostname	freenas.local
FreeNAS Build	FreeNAS-8.0.3-RELEASE-x64 (9395)
Platform	Intel(R) Core(TM) i3-2100T CPU @ 2.50GHz
Memory	8088MB
System Time	Tue Jan 10 08:38:12 2012
Uptime	8:38AM up 1:06, 0 users
Load Average	0.00, 0.00, 0.00
OS Version	FreeBSD 8.2-RELEASE-p5

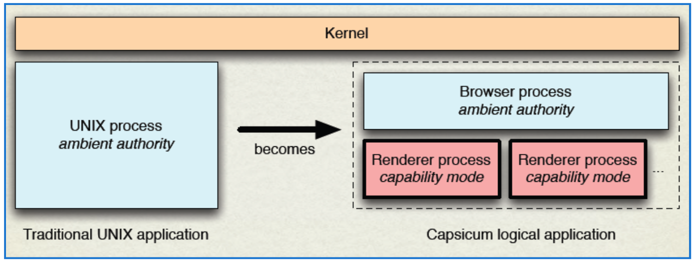


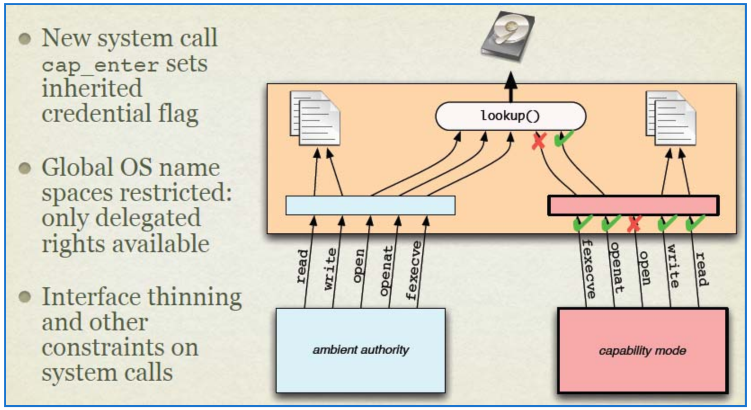
Edit	ID	Name	Serial	Identifier	Description	Transfer Mode
Edit Replace	2	ada1	JP2940HZ:	{uuid}5a83889 3c6c-11e1-af83-e06995777	Member of volume1 stripe	Auto
Edit Replace	3	ada2	JP2940HZ:	{uuid}5ae8db 11e1-af83-e06995777a82	Member of volume1 stripe	Auto
Edit Replace	4	ada3	JP2940HZ:	{uuid}cf28493 3c6c-11e1-af83-e06995777	Member of volume1 stripe	Auto
Edit Detach	1	ada0	JP2940HZ:	{uuid}5a2455f 3c6c-11e1-af83-e06995777	Member of volume1 stripe	Auto

BSD Cool Hot Stuff

- **Capsicum** – a method of *userland* sandboxing applications. Each process gets “capabilties” via the capsicum library that integrates new security features while remaining POSIX compliant . Can be done in very few lines of code. (Chromium ~100 LoC)

- **Capabilities** - refined file descriptors with fine-grained rights
- **Capability mode** - process sandboxes that deny access to global namespaces
- **Process descriptors** - capability-centric process ID replacement
- **Anonymous shared memory objects** - an extension to the POSIX shared memory API to support anonymous swap objects associated with file descriptors (capabilities)
- **rtld-elf-cap** - modified ELF run-time linker to construct sandboxed applications
- **libcapsicum** - library to create and use capabilities and sandboxed components
- **libuserangel** - library allowing sandboxed applications or components to interact with user angels, such as Power Boxes.
- **chromium-capsicum** - a version of Google's Chromium web browser that uses capability mode and capabilities to provide effective sandboxing of high-risk web page rendering.
- **tcpdump** was sandboxed with 8 lines of code





- New system call `cap_enter` sets inherited credential flag
- Global OS name spaces restricted: only delegated rights available
- Interface thinning and other constraints on system calls

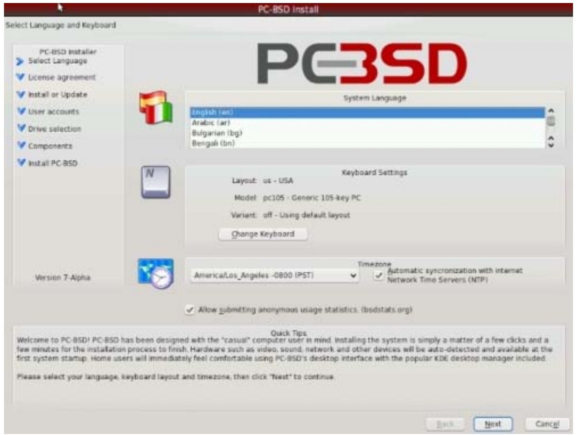
BSD Cool Hot Stuff

- Virtualization – a very quick summary.
  - **Jails** – lightweight, individual virtualized subsystems with their own processes, files and accounts (including root). Ideal for hosting environments. Each jail is a separate application environment on the host. Hosts with > 5,000 jails are common (max I've heard is over 60,000)
  - **VIMAGE** - Allows for complete networking independence between jails on a system, including giving each jail its own firewall, virtual network interfaces, rate limiting, routing tables, and IPSEC configuration. Available as a kernel compile option.
  - **Xen** – Hypervisor-based virtualized systems complete with virtualized hardware. Dom0 hosts with DomU guests. Both FreeBSD and NetBSD now support Dom0 functionality.
  - **VMWare** – Commercial virtualization technology. FreeBSD runs as a guest.
  - **Hyper-v** - Microsoft virtualization technology. FreeBSD enlightened IO drivers under development by Microsoft and NetApp. Beta available summer 2012.
  - **EC2** – FreeBSD now available on Amazon EC2 instance types.

    http://forums.freebsd.org/showthread.php?t=28650
  - Other virtualizations – QEMU, AQEMU, VirtualBox

BSD Cool Hot Stuff

- **PC-BSD** – developed by Kris Moore, released in 2006 with FreeBSD 6.0
  - Officially supported by iXsystems. Large scale deployments (over 5000 desktops) are being supported. Very active community.
  - Full graphical installer, supports ZFS and UFS; scriptable install engine
  - Full FreeBSD install with customized scripts for full desktop experience, including auto device management, wireless, etc.
  - Desktops - GNOME2, KDE4, LXDE, and XFCE4 supported; Awesome, Etoile,FVWM, i3, IceWM, Openbox, Spectrwm, Window Maker, or WindowLab unsupported, but available.
  - Portable Binary Installer – PBI for one-click package management.
  - App Café for new applications.
  - Common OSS support (Open Office, MySQl, Postgres, etc.)



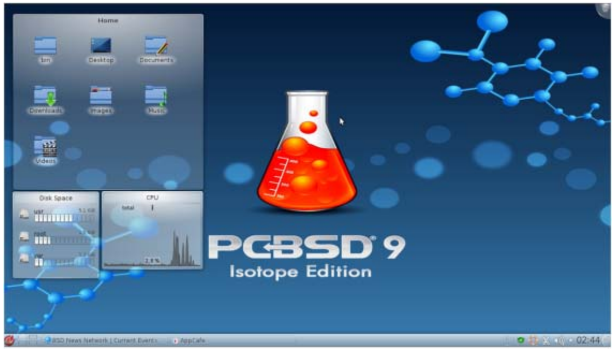
Layout: us - USA
Model: pc105 - Generic 105-key PC
Variant: off - Using default layout

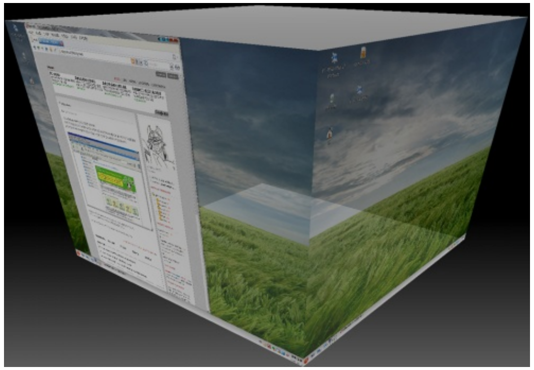
America/Los_Angeles -0800 (PST)
☑ Automatic syncronization with internet Network Time Servers (NTP)

☑ Allow submitting anonymous usage statistics. (bsdstats.org)

Welcome to PC-BSD! PC-BSD has been designed with the "casual" computer user in mind. Installing the system is simply a matter of a few clicks and a few minutes for the installation process to finish. Hardware such as video, sound, network and other devices will be auto-detected and available at the first system startup. Home users will immediately feel comfortable using PC-BSD's desktop interface with the popular KDE desktop manager included.

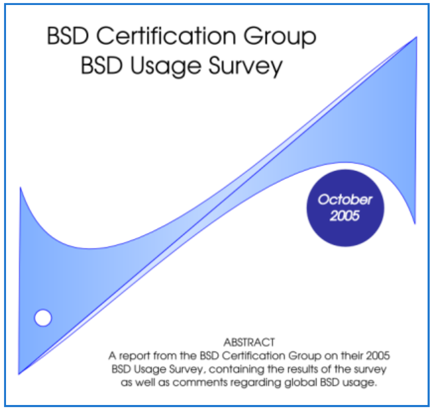
Please select your language, keyboard layout and timezone, then click "Next" to continue.





BSD Family Notes

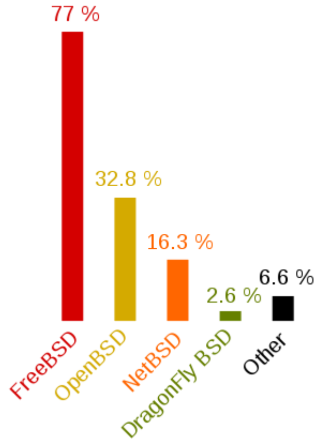
- Many people get upset with the “daemon” reference items - horns, trident, etc.
  - Best thing is to focus on the system and its capabilities, not the ‘packaging’.
  - It’s changed recently with new logos, and a lot wider audience. The four major project now have logos that are more acceptable to a wider variety of professional interests.
  - Making noise about it will likely cause the BSD community to roll eyes and ignore you. It’s been discussed, argued, incensed, flamed (!), vilified, maligned, disparaged, etc. countless times. You won’t change anyone’s mind- trust me.
- Many organizations use BSD internally, but are reluctant to publicize that fact. A survey by the BSDCG in 2005 revealed that many companies are in fact running BSD systems, including some very large installations.



BSD Certification Group
BSD Usage Survey

October 2005

ABSTRACT
A report from the BSD Certification Group on their 2005 BSD Usage Survey, containing the results of the survey as well as comments regarding global BSD usage.



BSD Certification

- The BSD Certification Group offers rigorous, psychometrically valid certification exams for BSD system administrators (**BSDA**, **BSDP**)
- Open since 2005, first exam released in 2008
- Computer based exam available in US and abroad via Schroeder Measurement Technologies
- Globally affordable
- www.bsdcertification.org – a 501(c)(6) organization
- Full Disclosure – I'm VP / Treasurer of the BSDCG

BSD Overview

Questions?

Thanks!

Commercial Uses of BSD - a partial list

- **Apple Inc.'s Darwin**, the core of Mac OS X and iOS; built on the XNU kernel (part Mach, part FreeBSD, part Apple-derived code) and a userland much of which comes from FreeBSD
- **Blue Coat Systems** network appliances
- **Calexium MailFountain** is an Email Center appliance based on FreeBSD 8.1
- **Borderware appliances** (firewall, VPN, Anti-SPAM, Web filter etc) is based on a FreeBSD kernel.
- **Check Point IPSO** security appliances
- **Citrix Systems Netscaler** Application Delivery Software is based on FreeBSD.
- **Coyote Point** GX-series Web Acceleration and Load Balancer appliances
- **Dell** (was EqualLogic) iSCSI SAN arrays
- **Halon Security H/OS** 1.3.X is based on FreeBSD 6.2, H/OS 1.4.X is based on FreeBSD 7.2, H/OS 2.X is based on FreeBSD 7.0
- **IronPort AsyncOS** is based on a FreeBSD kernel
    - AntiSpam
    - SenderBase
- **Isilon Systems' OneFS**, the operating system used on Isilon IQ-series clustered storage systems
- Juniper Networks **JUNOS**
    - JUNOS prior to 5.0 was based on FreeBSD 2.2.6.
    - JUNOS between 5.0 and 7.2 (inclusive) is based on FreeBSD 4.2.
    - JUNOS 7.3 and higher is based on FreeBSD 4.10.
    - JUNOS 8.5 is based on FreeBSD 6.1
    - JUNOS 10.0 is based on FreeBSD 7.
- **KACE Networks's KBOX** 1000 & 2000 Series Appliances and the Virtual KBOX Appliance
- **nCircle's IP360** security products use FreeBSD 6.x
- **McAfee SecurOS**, used in e.g. Firewall Enterprise (aka Sidewinder)
- **NetApp** filers based on Data ONTAP
- **Netasq** intrusion prevention appliances
- **COMP VPN gateways**, some of them certified by the Internal Security Agency for processing classified data
- **Panasas** parallel network storage systems
- **Panasonic** uses FreeBSD in their Viera TV receivers
- **Silicon Graphics International** uses FreeBSD in their MAID disk arrays, formerly manufactured by COPAN.
- **Sophos's** Email Appliance
- **Statseeker**, Network Monitoring Software
- **St. Bernard Software** iPrism web filter appliance
- **Symmetricom** Timing Solutions
- **VXworks**

pf Syntax Examples

Lists

```
block out on fxp0 from { 192.168.0.1, 10.5.32.6 } to any
```

gets expanded to:

```
  block out on fxp0 from 192.168.0.1 to any
  block out on fxp0 from 10.5.32.6 to any
```

Multiple lists can be specified within a rule:

```
  match in on fxp0 proto tcp to port { 22 80 } rdr-to 192.168.0.6
  block out on fxp0 proto { tcp udp } from { 192.168.0.1, \
     10.5.32.6 } to any port { ssh telnet }
```

Note that the commas between list items are optional.

Lists can also contain nested lists:

```
  trusted = "{ 192.168.1.2 192.168.5.36 }"
  pass in inet proto tcp from { 10.10.0.0/24 $trusted } to port 22
```

Macros

```
  ext_if = "fxp0"
  block in on $ext_if from any to any
```

Macros can also expand to lists, such as:

```
  friends = "{ 192.168.1.1, 10.0.2.5, 192.168.43.53 }"
```

Macros can be defined recursively. Since macros are not expanded within quotes the following syntax must be used:

```
  host1 = "192.168.1.1"
  host2 = "192.168.1.2"
  all_hosts = "{" $host1 $host2 "}"
```

Tables

```
  table <goodguys> { 192.0.2.0/24 }
  table <rfc1918> const { 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8 }
  table <spammers> persist

  block in on fxp0 from { <rfc1918>, <spammers> } to any
  pass  in on fxp0 from <goodguys> to any
```

Address Matching

An address lookup against a table will return the most narrowly matching entry. This allows for the creation of tables such as:

```
  table <goodguys> { 172.16.0.0/16, !172.16.1.0/24, 172.16.1.100 }
  block in on dc0
  pass  in on dc0 from <goodguys>
```

Table Maniuplation

Tables can be manipulated on the fly by using pfctl(8). For instance, to add entries to the <spammers> table created above:

```
  # pfctl -t spammers -T add 218.70.0.0/16
```

This will also create the <spammers> table if it doesn't already exist.
To list the addresses in a table:

```
  # pfctl -t spammers -T show
```

The -v argument can also be used with -Tshow to display statistics for each table entry. To remove addresses from a table:

```
  # pfctl -t spammers -T delete 218.70.0.0/16
```

Complete pf Example

Rule Syntax

The general, highly simplified syntax for filter rules is:

```
action [direction] [log] [quick] [on interface] [af] [proto protocol] \
   [from src_addr [port src_port]] [to dst_addr [port dst_port]] \
   [flags tcp_flags] [state]
```

```
# Pass traffic in on dc0 from the local network, 192.168.0.0/24,
# to the OpenBSD machine's IP address 192.168.0.1.
#  Also, pass the return traffic out on dc0.
pass in  on dc0 from 192.168.0.0/24 to 192.168.0.1
pass out on dc0 from 192.168.0.1 to 192.168.0.0/24
```

```
# Pass TCP traffic in on fxp0 to the web server running on the
# OpenBSD machine. The interface name, fxp0, is used as the
# destination address so that packets will only match this rule if
# they're destined for the OpenBSD machine.
pass in on fxp0 proto tcp from any to fxp0 port www
```

Using the 'quick' keyword:

```
block in quick on fxp0 proto tcp to port ssh
pass  in all
```

```
ext_if  = "fxp0"
int_if  = "dc0"
lan_net = "192.168.0.0/24"

# table containing all IP addresses assigned to the firewall
table <firewall> const { self }

# don't filter on the loopback interface
set skip on lo0

# scrub incoming packets
match in all scrub (no-df)

# setup a default deny policy
block all

# only allow ssh connections from the local network if it's from the
# trusted computer, 192.168.0.15. use "block return" so that a TCP RST is
# sent to close blocked connections right away. use "quick" so that this
# rule is not overridden by the "pass" rules below.
block return in quick on $int_if proto tcp from ! 192.168.0.15 to $int_if port ssh

# pass all traffic to and from the local network.
# these rules will create state entries due to the default
# "keep state" option which will automatically be applied.
pass in  on $int_if from $lan_net
pass out on $int_if to $lan_net

# pass tcp, udp, and icmp out on the external (Internet) interface.
# tcp connections will be modulated, udp/icmp will be tracked statefully.
pass out on $ext_if proto { tcp udp icmp } all modulate state

# allow ssh connections in on the external interface as long as they're
# NOT destined for the firewall (i.e., they're destined for a machine on
# the local network). log the initial packet so that we can later tell
# who is trying to connect. use the tcp syn proxy to proxy the connection.
# the default flags "S/SA" will automatically be applied to the rule by PF.
pass in log on $ext_if proto tcp to ! <firewall>  port ssh synproxy state
```

Credits and Sources

- Sources:
  - Wikipedia
    - Ritchie, Thompson Award
    - Unix Family Tree
    - PF Feature Chart
    - BSD Comparison Chart
    - Commercial Uses List
  - Usenix 25th Anniversary Card Deck
    - Pictures of CSRG members (except Leffler)
  - pfSense Website
  - iXsystems Website
  - Dr. Robert Watson website
  - BSDCG Website
  - Images.google.com

- The BSD Certification Group Inc.`s logo has been designed by FJZone.org as part of its commitment to philanthropic ventures worldwide. Copyright 2006 by the BSD Certification Group Inc.
- Fred Artwork Copyright 2005 by DragonFly BSD.
- The mark FreeBSD is a registered trademark of The FreeBSD Foundation and is used by The BSD Certification Group, Inc. with the permission of The FreeBSD Foundation.
- The FreeBSD Logo is a trademark of The FreeBSD Foundation and is used by The BSD Certification Group, Inc. with the permission of The FreeBSD Foundation.
- NetBSD is a registered trademark of The NetBSD Foundation, Inc. The NetBSD Logo Copyright 2004 by The NetBSD Foundation, Inc.
- Puffy Artwork Copyright 2004 by OpenBSD.

- Use of the above names, trademarks, logos, and artwork does not imply endorsement of this presentation program by their respective owners.