Packet Filters and Firewalls

Beside the packet filter 'pf' from OpenBSD, FreeBSD also has 'ipfw' (still used in MacOS X) and 'IPFilter' as firewalls.

In conjunction with pfsync and CARP pf opens up unique possibilities to protect a network.

With comparably small effort networks can be connected to the Internet in a completely redundant and connection-oriented way. If you want redundant firewalls state information has to be available to all nodes in the setup otherwise connectivity cannot be maintained if one node fails. The pfsync-protocol was developed to exchange this connection-state information.

With CARP it is possible, to put it in simple terms, to 'hide' a number of machines behind one single IP address. If one machine fails, one of the other machines takes over. In addition to this it is also possible to balance the load over all available nodes and maintain connectivity even when multiple machines fail. CARP has full IPv6 support and with its integrated encryption is protected against attacks which VRRP (Virtual Router Redundancy Protocol) isn't.

Jails

'chroot' is one of the oldest security mechanisms known, it limits access to just a part of the file system. FreeBSD of course also has chroot. While it's suitable for simple tasks it has its limits. That's why FreeBSD implemented Jails, which are basically virtual environments (clients) on a file system completely isolated from the host system.

While it's possible for root to break out of a chroot-environment, this is not possible in a Jail. Each Jail has its own IP address, its own system configuration and can execute its own applications.

A Jail shares the resources with its host system. Changes in the host can affect the Jail, but not vice-versa. It's a special form of sandboxing. Jails can also serve as system to keep order and to do backups. If you need more information you can visit: http://docs.FreeBSD.org/44doc/papers/jail/jail.html and http://en.wikipedia.org/wiki/FreeBSD_jail

Disk Encryption

FreeBSD supports complete encryption of partitions and Swap via geli(8) and gdbe(8) including separate storage of the keys on USB sticks etc. The possibilities are many and we can't handle them all in this flyer. Please see the additional Geom and Encryption flyer.

Security Event Auditing (TrustedBSD)

For special high-security applications the FreeBSD auditing is available starting with FreeBSD 6.2. It has a Solaris-compatible audit framework called OpenBSM. Warning: The audit facility in FreeBSD 6.2 is experimental, and production deployment should occur only after careful consideration of the risks of deploying experimental software. Please have a look at: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/audit.html

Further Information

http://www.FreeBSD.org/

http://www.FreeBSD.org/doc/en/books/handbook/

http://lists.FreeBSD.org/mailman/listinfo

http://www.BSDForums.org/

We cannot possibly list all security features of FreeBSD in detail on 2 pages. Further flyers will follow, e.g. on firewalls. If you have any comment, critique or improvement to make, please don't hesitate and send a mail to Daniel.Seuffert@allBSD.de. Thanks a lot!

FreeBSD: The Power To Serve

FreeBSD

Security without compromise

Like all BSDs, FreeBSD too is known for its uncompromising stance on security. To maintain this an organizational structure has evolved over the years which is constantly being improved and extended.

The Security Team And Its Structure

All processes and structures relevant to handling security issues are publicly visible, please see http://www.FreeBSD.org/security The FreeBSD Security Officer carries the responsibility for the organization. He is assisted by the Security Officer Emeritus, the Deputy Security Officer, a member of the Core Team and members of the security team. All known and verified vulnerabilities are published in a public database at http://www.vuxml.org/FreeBSD

Older releases of FreeBSD are of course still maintained with security patches:

Early Adopter: Releases from -CURRENT (the development branch) are supported for a minimum of 6 months after publishing.

Normal: Releases from -STABLE (the production branch) are supported for a minimum of 12 months, mostly longer.

Extended: Select releases are supported for a minimum of 24 months, mostly longer.

Where Do I Find Information As A User?

FreeBSD has excellent documentation, the handbook of course also describes all security mechanisms, see: http://www.FreeBSD.org/doc/en/books/handbook/security.html

There you will find help on installation, usage and maintenance of OpenSSL, Kerberos, VPNs with IPSec, OpenSSH, One-Time Passwords, PAM, TCP-Wrappers, Access Control Lists (ACLs) for file systems, process accounting etc. As you see with this short list there are many ways you can protect your system.

Security Advisories

Security vulnerabilities may come up at any time. To inform the users the FreeBSD Security Team sends so called Security Advisories (SA) per Email or publishes them on the FreeBSD project website. There are no SAs for applications in Ports. In the SAs for the operating system the problem is precisely described and clear instructions on obtaining the patch and closing the hole are given.

The number of SAs varies, but there are approximately 30-40 per year. However not everyone is affected by each SA, although you should read each and check how it applies to you. Make sure you are subscribed to the announce- or security-mailinglist or use the RSS feed.

Binary Updates With FreeBSD Update

Many users and administrators of server farms prefer binary updates which are easy to use and don't require a recompile of the whole or just parts of the operating system. F**reebsd-Update** enables a semi-automatic installation of patches similar to the 'Windows Update' mechanism by Microsoft. If you like you can use a cronjob to automatically apply updates and all updates can be versed if needed. Updates are available for all new FreeBSD releases. Please refer to http://www.daemonology.net/freebsd-update or the manual page for more information.

Freebsd-Update is now in the FreeBSD base and no longer a separate Port (starting with FreeBSD 6.2). It supports only the GENERIC kernel at the moment but more improvements are underway.

Security Profiles, Securelevels, File System Security Flags

A Security Profile is a collection of settings that controls which programs and options are active. One principle of BSD is to only have the programs running that you really need. After you installed FreeBSD you activate or deactivate all programs in '/etc/rc.conf'. Please refer to http://www.FreeBSD.org/doc/en/books/handbook/install-post.html for more information.

Securelevels are another security mechanism, it can be set with the variable 'kern.securelevel' via 'sysctl'. After setting the Securelevel to "1" all writing operations to raw devices are forbidden and special chflags options like schg are enforced. schg is the so-called 'system immutable flag', even more strict than sappnd, it prevents all writes to files. FreeBSD routinely uses this on a number of configuration files and the kernel. Securelevels should be used by experienced users only. Securelevels have nothing to do with 'runlevels' known from other operating systems like Linux.

How Do I Find Vulnerable Applications?

The FreeBSD project cannot maintain and distribute SAs for all 16'000 Ports. But there is a way to keep an eye on the programs too. The program Portaudit was explicitly developed for this.

The Port '../security/portaudit' queries a database maintained by the FreeBSD Security Team and the Port Maintainers for any known vulnerability. During installation the configuration files of periodic(8) are updated so Portaudit can enter its findings in the daily security report. You only have to make sure you read the reports when they are sent to 'root'.

After installation of the portaudit-database you can check your installed applications at any time. If there is a known vulnerability in an installed program, Portaudit prints a URL with information on the vulnerability. Portaudit, combined with Portupgrade and Portsnap is an extremely valuable tool to maintain your system and keep it safe.

Installing Additional Programs

FreeBSD has more than 16'000 Ports available for installation. At http://www.freshports.org/security you can find over 600 security-related programs, from Intrusion Detections Systems, to Snort, GnuPG, Aide and Honeyd. The choice is huge you will most probably be able to find a program suitable to your task.

Code Quality: Audits, Coverity, Styleguides

FreeBSD is renowned for its high code quality and tries to improve on it constantly. There is a public CVS that can be accessed with a web-interface and many people check new commits to the code base on a daily basis. FreeBSD is a professional organization and each committer has to follow stringent guidelines: http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/developers-handbook/secure.html

To support this process FreeBSD also uses the Coverity Prevent Analysis Tool which checks every change to the code base every day. It finds security problems in C and C++ code in over 30 categories. Additionally constant regression tests are done. A dedicated infrastructure is currently being set up to do daily and automated regression tests. And there is also a dedicated test environment to do Stress Tests to check the stability of the code. Please refer to http://www.holm.cc/stress for more information.

Every user of FreeBSD can report bugs and problems with through an easy interface. As always this information is also public and can be seen here: http://www.FreeBSD.org/cgi/query-pr-summary.cgi or http://www.FreeBSD.org/prstats/index.html