Some Technical Details

The implementation of Jails is not restricted to userland, it's an integral part of the kernel.

A Jail requires approximately 140 MB disk space, but can of course be more depending on what you put in it. On the other hand you can also strip it down and delete anything you don't need to run the actual jailed service.

From the system administrator's point of view a Jail is handled like any other system. You can log into it via SSH. From the point of view of the host system a Jail is only directory with a complete minimal FreeBSD installation on which you have access from the host system. To work with the Jail you don't have to log into Jail. With 'jexec' you can start and stop daemons and programs, just look at running processes or which users are logged in.

In the host system's process list every Jail-process is marked with a 'J'. That way you can immediately see which processes are running in Jails on the host.

A Jail is subject to certain restrictions which you can influence via the host system's rc.conf(5) and sysctl(8). Normally raw sockets (eg. for ping and traceroute) are not allowed, that can be changed however, as can setting the Jail's host name from within the Jail. One of the biggest restrictions with Jails though is that only one IP address per Jail is allowed, and the address has to be set with netmask '/32'.

A lot of improvements are in the making as of lately: ipv6-support, more than 1 IP per jail and resource limits.

The success has led to a project named sysjail for NetBSD; OpenBSD and MirOS, please see http://sysjail.bsd.lv/
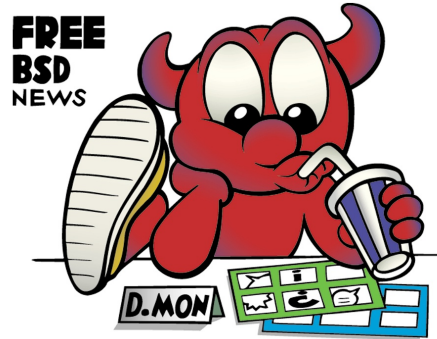
FreeBSD
Jails

Further Information

With this short overview we hope we have been able to show you how FreeBSD Jails can work for you too.

The Wikipedia also has an article on Jails:

http://en.wikipedia.org/wiki/FreeBSD_jail

The manual page for Jails is very important:

man jail

If you're more interested in FreeBSD's security features, please see following flyers too:

Firewall and Security

General information on FreeBSD is to be found here:

http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/

What are Jails?

You can describe a Jail as being a chroot on steroids. But a Jail is much more than a chroot-environment. A Jail is its own complete FreeBSD system with its own IP address within a FreeBSD host system.

A Jail is restricted in its processes and child processes. Processes in a Jail cannot access processes on the host system. The hardware is not emulated like with vmware. There is also no special kernel like with XEN. A Jail shares the resources with the host system, without affecting the host system if there are changes in a Jail. This is the reason for the superior performance compared to other virtualization techniques, especially when multiple virtual instances are running at the same time.

FreeBSD Jails belong to the base system and can increase the security of your system significantly. Instead of a whole system you can also just Jail a single process. A Jail is a complete system, that means you can install the same software as you would on a normal FreeBSD system.

Server Processes In Jails

Especially daemons like DNS, HTTP, SMTP, POP3, IMAP, FTP and many other ones had vulnerabilities which a hacker could use to access gain access to a system. Even when diligently patching a system and running it behind a firewall, a risk remains. To minimize this risk it's advisable to lock a daemon in a Jail.

Whether you put all daemons in one Jail or each daemon in its own is up to you. Even after a Jail has been broken into, your host system remains intact.

Security through Jails

When an intruder breaks into a system, you cannot be sure only the software that served as entry point was exploited. Maybe a backdoor was installed. Maybe subtle errors and misconfigurations were introduced. What is the downtime you have to plan for after a successful break-in? Time is money and with Jails you can save a lot of each. When your webserver gets compromised you don't have to waste time doing a forensic examination of the system.

Just stop the Jail, delete it, and copy the Backup-Jail to the webserver's place. This would not take longer than a minute. And at your leisure you could then do a forensic examination of how your server was compromised. kompromittiert, so suchen Sie nicht lange nach veränderten Dateien.

Your Server as Fort Knox

It's safe to say that nothing is secure, even if makers of various firewall solutions tell you different. Firewalls are the first barrier, the operating system as such is the second barrier, Jails form the third barrier and if you use Security Flags you even get a safe without a door. A compromised system is bad enough, within a Jail it's not that bad. But it can get really annoying if the intruder changes or destroys data, or deletes his traces in the log files. He needs root access for this and he may get that in the Jail.

You can stop that with Security Flags and make selected files read-only or append-only. And with Secure Levels you can even stop an intruder from modifying firewall rules or loading and unloading kernel modules or opening disks for write access. More info to Security Flags and Secure Levels can be found in the appropriate flyer.

A Jail Can Do More

Obviously a Jail is primarily use for security reasons. But there are also other advantages:

- You can run multiple virtual servers with different daemons on one server. You can run a whole DMZ on one server.

- Daemons are often administered by multiple administrators. Either all have root access or you need a well-configured sudo-file. If you lock all daemons in Jails you can give each administrator his own Jail.

- You need a test environment for your developers? Why install a whole server just for that, with it's own hardware even? Simply install a Jail for your developers where they have root access.

- If you want to offer root-shells to customers you can do this safely with Jails. You don't need to buy new hardware and you can easily automate Jail setups.

- If you're teaching you can give each student her own server in a Jail. She can be root in it and do anything you allow. When class is over you don't need to reinstall the server, simply delete the Jails.

- As you just saw there are many more things you can do with Jails than just improve security. Above all you can save money just by using your existing hardware to its full extent.