Notes - compiling OpenBSD kernel

OpenBSD -
an operating system secure by default.

The OpenBSD project produces a **FREE**, multi-platform 4.4BSD-based UNIX-like operating system. The operating system emphasizes portability, standardization, correctness, proactive security [1] and integrated cryptography [2]. OpenBSD supports binary emulation of most programs from SVR4 (Solaris), FreeBSD, Linux, BSD/OS, SunOS and HP-UX.

These note are historical and here for reference by request, please review the official OpenBSD documentation. [3]

Notes to consider when compiling kernel

- **For using SAMBA** some administrators recommend setting option "NMBCLUSTERS=8192" when recompiling kernel to improve performance for the packet acknowledgement used in file sharing (increasing to 4096 may be sufficient if kernel errors arise), the generic kernel is said to be more configured to optimized for streaming based protocol.
"option NMBCLUSTERS=value, Size of kernel mbuf cluster map, mb_map, in CLBYTES-sized logical pages. Default on most ports is 256 (512 with "option GATEWAY"). See /usr/include/machine/param.h for exact default information. Increase this value if "mb_map full" messages appear."

- **To take advantage of some kernel options that may improve performance** may want to verify or add the following options:
option FFS_SOFTUPDATES
option GATEWAY, sets several options useful to be gateway
option BUFCACHEPERCENT=integer Percentage of RAM to use as a file system buffer. It defaults to 5, some servers in examples use 20 or 50 to use the memory that is usually idle.
option DUMMY_NOPS, General speed hack that was not enabled in the generic kernel because it may not work in a minority of platforms.
option UVM, Advanced Virtual Memory system. Speeds up a machine when swapping.
option MFS Memory File System. Can be used to create RAMDISK partitions for extremely fast data access.
Increasing NKMEMPAGES has been said to be a bad idea.

- **Perhaps remove kernel references not used for hardware and software that you will not be needing.** For example most internet servers do not need the kernel to support I386_CPU, I486_CPU, GPL_MATH_EMULATE for processor, PCMCIAVERBOSE and other VERBOSE not needed,.pcmcia, usb, multiport serial interfaces, sound cards, may not need SCSI if not using, FM & TV receivers, nor wireless devices. Just be careful to get related lines commented or deleted in the configuration file. The thought here is "less is more' but look out for removing something needed...
A trick you could do to determine devices you may want to keep is to save and review the /var/run/dmesg.boot created by a GENERIC kernel on the server you are hardening the kernel for.

- **You may desire to disable kernel options specified in an included file.** For example if you are never going to use NFS, you may not need the options NFSCLIENT nor NFSSERVER. But those options are included by the GENERIC included kernel. Rather than editing the GENERIC, which is not a good idea, you can use the "rmoption" like this

  rmoption NFSCLIENT
  rmoption NFSSERVER

- **Disabling stuff like SYSVSHM might help for resources, but read up on it first** especially if using applications like postgres that use shared memory:
  "A large Postgres installation can quickly hit various operating system resource limits. [4] Shared memory and semaphores are collectively referred to as "System V IPC" (together with message queues, which are not relevant for Postgres)... Almost all modern operating systems provide these features, but not all of them have them turned on or sufficiently sized by default, especially systems with BSD heritage....
  The options `SYSVSHM` and `SYSVSEM` need to be enabled when the kernel is compiled. (They are by default.) The maximum size of shared memory is determined by the option `SHMMAXPGS`...."

- **To further harden** the system some administrators suggest the following:
  - Note that the file you normally edit contains and include. Follow the include and note you may want to either edit or copy the file to include to a different name and change accordingly the include statement. This way you can edit options and features in the included file also.
  - DDB (kernel debugger), KTRACE (system call tracing) and KMEMSTATS (memory allocation statistics) options probably can be turned off
  - Another option than may harden if removed is the option LKM, loadable kernel modules which "allow the system administrator to dynamically add and remove functionality from a running system"
  - May want to disable file system options that will not be used like EXT2FS if not using linux, UNION, and NFS can be turned off. Both NFS client and server support if not used could be disabled to reduce risks. Some of the file system commands may require disabling in the include file like NULLFS, PORTAL, PROCFS and UMAPFS.

  - Some may want to consider disabling support for IPv6 and IPsec options along with the related PULL_DOWN and CRYPTO options and enc pseudo device if your server will not need them. We think you should not do this because you may introduce new holes while trying to fix something that is not broke. Note that in OpenBSD 3.0, disabling IPv6 will cause sendmail to fail when it's started with the default configuration. To allow sendmail to run without IPv6 two lines must be removed from the active localhost.cf file in /etc/mail. They are the two lines that start with "O DaemonPortOptions=Family=inet6,". The quick way is to comment them out with a # sign at the begining of the line. When making a custom sendmail configuration file will need to edit the mc file accordingly if IPv6 support is disabled for the kernel.

- There are other options that may be considered for encrypting the swap space and making the actual file systems harder to get...

Links to web pages related to recompiling the OpenBSD kernel

- Section 5.0 of the main OpenBSD FAQ discusses kernel configuration
  http://www.openbsd.org/faq/faq5.html [5]

- =8> nomoa.com/bsd OpenBSD - kernel docs
  http://nomoa.com/bsd/kernel.htm [6]
  =8> nomoa.com/bsd OpenBSD - optimizing kernel for SAMBA
  http://nomoa.com/bsd/samba.htm#kernel [7]

- GeodSoft Website Consulting; Hardening OpenBSD Internet Servers Building a Custom Kernel
  http://geodsoft.com/howto/harden/OpenBSD/kernel.htm [8] |

- O'Reilly notes on OpenBSD Kernel Compilation and Optimization
  http://www.onlamp.com/pub/a/bsd/2000/10/31/OpenBSD.html [9]

Links
[1] http://www.openbsd.org/security.html
[2] http://www.openbsd.org/crypto.html
[3] http://www.openbsd.org/faq/faq5.html#BldKernel
[4] http://www.postgresql.org/idocs/index.php?kernel-resources.html
[5] http://www.openbsd.org/faq/faq5.html
[6] http://nomoa.com/bsd/kernel.htm
[7] http://nomoa.com/bsd/samba.htm#kernel
[8] http://geodsoft.com/howto/harden/OpenBSD/kernel.htm
[9] http://www.onlamp.com/pub/a/bsd/2000/10/31/OpenBSD.html