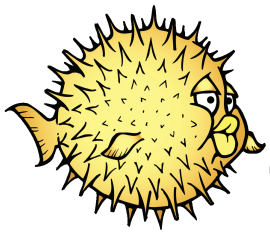
Xenocara - integrating X.Org in OpenBSD

Matthieu Herrb

OpenBSD/X.Org



OpenBSD

FOSDEM February 23, 2007

Agenda

Agenda

OpenBSD

Remainder:

- OpenBSD is a *free* multi-platform OS based on BSD 4.4.
- Focus on usability, portability, correctness and proactive security.
- One release every six months.
  Current version is 4.0, released Dec, 1.
- Kernel, user-land & docs maintained by the same people.
- Base system includes the X window system with a minimal set of applications (fvwm, xterm, etc.)
- Third party applications (inc. KDE, Gnome, OpenOffice) available through the ports mechanism.

Features

Strong networking capabilities:

- the PF packet filter
- routing daemons: OpenBPd, OpenOSPFd, ripd
- IPSec, IPv6
- high availability: CARP, pfsync, sasyncd, hoststated
- spamd

Hardware support:

- wireless adapters,
- disk controllers, including RAID management,
- ethernet cards,
- graphics cards.

OpenBSD ports

Goal: to provide *binary packages* of 3rd party applications, built from their source with patches to customise/fit them better on OpenBSD.

The ports *source tree* contains:

- references to the original sources and how to fetch them,
- a human readable description of the port,
- dependencies data (including shared lib revisions),
- a **packing list** which describes what's installed,
- a **make** based build infrastructure that controls:
  - fetching the sources and patching,
  - configuring/compiling with the provided tools,
  - installing into a temporary location and building the binary package.
  - installing the binary package into its final destination (`/usr/local`).

Agenda

X in OpenBSD

X has been integrated in OpenBSD's source for a long time (XFree86 3.3, then 4.x and now X.Org 6.8 and 6.9).
OpenBSD ships the applications in the old monolith tree plus some add-ons:

- fvwm, wm2: alternatives to twm
- ssh-ask-pass, xidle, xlock, xsystrace: security
- xvctl
- ws input driver

The X server supports 10 architectures (alpha, amd64, hp300, i386, mac68k, macppc, sparc, sparc64, vax and zaurus).

X sources

Maintained in the OpenBSD CVS repository.

Based on released versions, plus OpenBSD-specific patches :

- privilege separation,
- support for some legacy architectures,
- bug fixes not yet integrated or released upstream.

Merging as many local changes as possible back to X.Org.

X.Org modular tree

Several questions appeared with the modular tree:

- make sure that autotools based configuration also works on *BSD,
- how to manage dependencies between the 300+ modules?
- how to integrate 3rd party sources that are no more in X.Org?
- how to cope with individual releases of modules?
- what to do with the CVS repository?
- where to install the binaries?

The Xenocara build system

Xenocara aims at answering those questions:

- X will not move to the ports tree,
- Re-use ideas from ports to drive the autotools builds with make and adapt libtool's behaviour,
- Xenocara is much simpler than ports though.
  - `bsd.xorg.mk` is < 200 lines
  - `Makefile.bsd-wrapper` are mostly one-liners
  - dependencies handled by explicit ordering of make targets.
- New directory in the OpenBSD CVS repository: `xenocara`.
- Still installing everything in `/usr/X11R6`

In the process, a BSD-licensed implementation of `pkg-config` in perl was developed (Chris Kuethe & Marc Espie) and has replaced the GNU version in the ports tree.

Schedule

Xenocara was able to do a full build of X.Org for the 1st time on the way back from fosdem last year.

Since then, lots of changes and fixes have happened.

Initial plan was to switch to xenocara (X.Org 7.2 based) for 4.1, but X.Org 7.2 got delayed and some developers were uncomfortable with switching too late in the release cycle.

The switch will occur just after code unfreeze in april

Todo list

- Fix remaining problems in the ports tree (main offenders: FreeType 2.2.x & pkg-config files)
- Create branches in copies of X.Org git repositories for OpenBSD local changes. `git.xenocara.org` had a start at this.
- Merge more local changes back to X.Org
- Repair X.Org -current build on BSD (broken by Daniel's work on input)
- Get kdrive working with wscons input drivers and wsfb framebuffer (useful for Legacy architectures)
- wscons input for dmx
- resume work on security audits of the code
- DRI
- ...

$\longrightarrow$ the list is long, help is welcome.

Agenda

Not all the X World is GNU/Linux

A lot of work has been done for the modular tree by X.Org developers. Most of this work is nowadays done on Linux.

Exceptions: Solaris (Alan Coopersmith), *BSD (Jeremy C. Reed, & Matthieu Herrb), SCO (Kean Johnston), ...

Windows (Cygwin/X) and Mac OS X/Darwin support ?

The MIT/X11 license is **very important** for many non-GNU/Linux users of X.

Support of legacy hardware (1-8 bpp) is also important for some OpenBSD developers & users.

Guidelines for multi-platform X development

- The build system of X.Org modules should only depend on the tools already listed in ModularDevelopper's guide. (XCB causes us a problem...)
- Don't rely on `sh` being bash or `make` being GNU-make.
- Don't assume glibc or GNU userland run-time environment
- OpenBSD and others don't support DRI yet: please test that drivers build without DRI (or drop this option completely)
- Pay more attention to API changes and shared library revisions.

About automake/autoconf

Autoconf philosophy: test for features instead of specific operating systems/versions: is good. Helps to reduce `#ifdef` mazes.

X sources still need lots of work in this area

But there are things that cannot be tested easily by an autoconf test: thus X.Org new configure scripts are still full of os-specific tests, and they don't always work as intended on the targeted platform.

A bit painful to debug and fix.

Automake in `xserver` takes $>$ 1 hour alone on a 600 MHz arm CPU.

Not easy to avoid depending on installed automake/autoconf...

Libtool hell

Libtool is a necessary evilness but:

- have to reconfigure it for each lib (slow even with autoconf cache)
- too complicated, makes errors difficult to spot
- ... insert you favourite rant here...

Fortunately, using libtool from OpenBSD's ports tree gives us some benefit:

- control over shared libs revision numbers
- bug fixes

Agenda

Benefits of OpenBSD security features

Running X applications on OpenBSD automatically benefits of security features that are standard in OpenBSD :

- stack protection via Propolice
- W^X on supported platforms
- shared libs address space randomisation

(See Theo's papers on attacks mitigation techniques for details)

Privilege separation in the X server

Limit the impact of malicious code execution in the X server by applying privilege separation here too:

- Initialise all things that need root privileges as early as possible (before parsing the configuration file)
- fork and have the main process change (definitively) it's uid to a non-privileged one.
- let the (privileged) child handle the few things that need root after initialisation, with lots of tests and restrictions, to avoid abuse.

Works great, but the unprivileged process still has full access to the hardware on most platforms.

X vs Kernel security

Modern systems make a difference between root (uid==0 in userland) privilege and kernel privileges:

- BSD securelevel > 0
- SELinux and other MAC frameworks
- ...

The X server having full control on the hardware via I/O ports and physical memory access breaks these models (see Loic Duflot's paper at CanSecWest'06).

Something ought to be done about that...

Solutions

- kernel-aided libpciaccess to filter PCI config space writes
- **vesafb**: unaccelerated dumb framebuffer based on VESA $\geq$ 2.x BIOS. (different from X's *vesa* driver)
- new driver model where all initialisation (including mode setting) is done in the kernel drivers (similar to Linux fbdev).
- KGI? (what's the good interface for acceleration?)
- EGL?

OpenBSD currently implements vesafb.

Agenda

Short future - OpenBSD 4.1

- Still based on X.Org 6.9
- vesafb + wsfb for an unprivileged X server

Longer term - OpenBSD 4.2 and beyond

- Switch to Xenocara (X.Org 7.2) just after 4.1 release
- Fix ports problems wrt modular X
- More in-kernel support for graphics cards
- DRI?
- nice graphics by Ty Semaka...

Questions ?