OpenBSD-Port: KedPM

Jörg Zinke

Entwicklungsprozesse in Open-Source-Projekten

13. Januar 2006

Berkeley-Software-Distributionen

- die 3 grossen heute:
  - FreeBSD (http://www.freebsd.org/)
  - NetBSD (http://www.netbsd.org/)
  - OpenBSD (http://www.openbsd.org/)
- man versteht sich nicht unbedingt als Konkurrenz sondern arbeitet in vielen Bereichen zusammen, übernimmt Code (insbesondere Gerätetreiber)
- weitere Derivate: ArchBSD, ClosedBSD, DesktopBSD, **DragonFly BSD**, Eclipse/BSD, **EkkoBSD**, emBSD, Firefly BSD, **FreeSBIE**, GoBSD, GuLIC-BSD, HpBSD, MicroBSD, **MirOS BSD**, NetBoz, PC-BSD, **PicoBSD**, SecureBSD, theWall, TrustedBSD,...

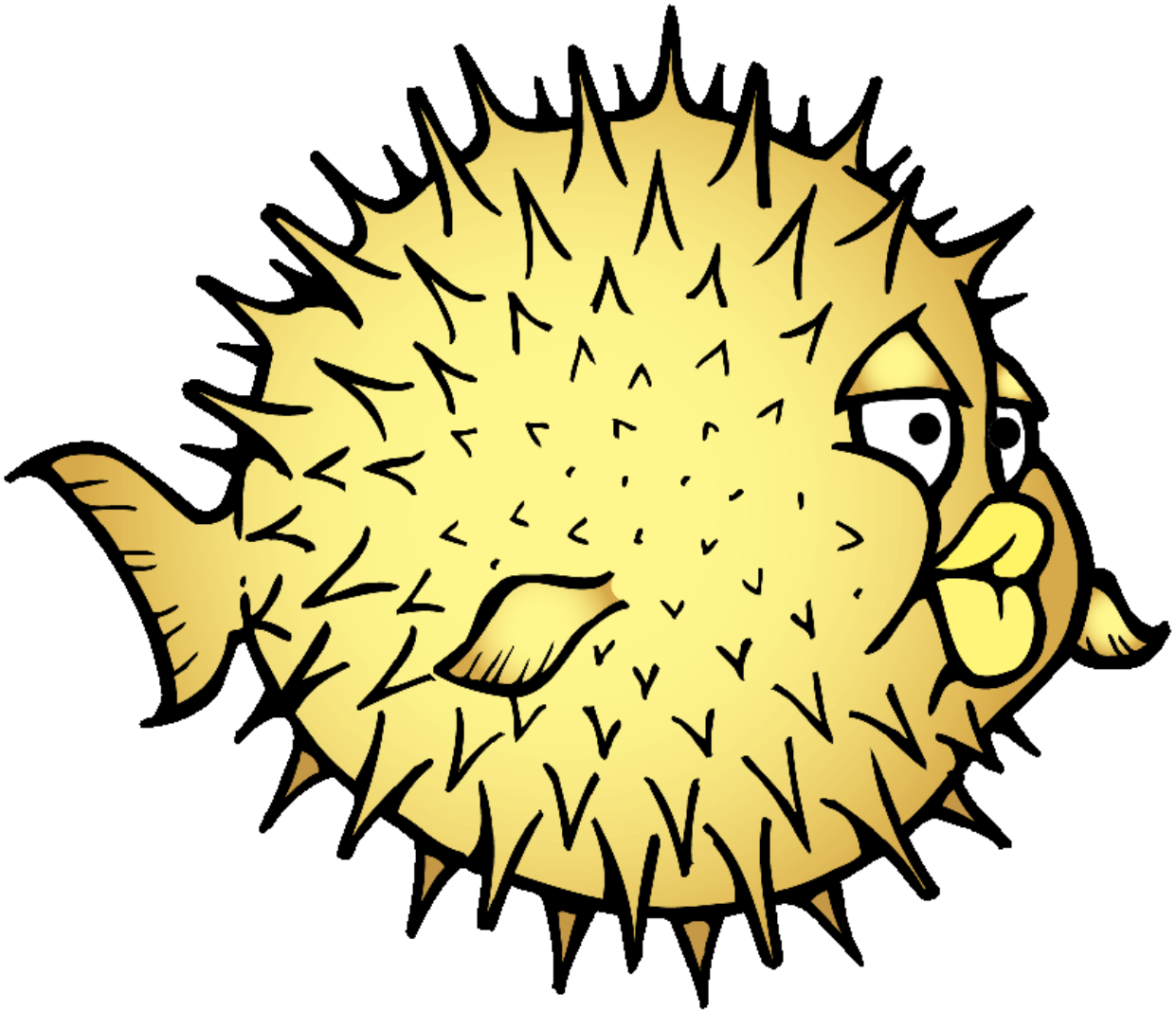
FreeBSD - *“The Power to serve.”*

- sicherlich bekanntestes BSD, sehr verbreitetes und modernes System
- setzt sich selbst als Ziel besonders einfach installierbar zu sein
- sowohl auf Desktops als auch auf Servern sehr verbreitet (ähnlich Linux)
- für Umsteiger von Linux sicherlich am empfehlenswertesten (grosse Auswahl an portierter Software)
- Mac OS X basiert im Kern auf FreeBSD

NetBSD - *“Of course it runs NetBSD.”*

- läuft auf nahezu jeder nur erdenklichen Hardware mit unterschiedlichsten CPU-Typen
- Auszug aus Liste der (aktuell **54!**) unterstützten Platformen: atari, amiga, playstation2, dreamcast, toaster,...
- ebenfalls ein sehr modernes System mit grosser Entwicklergemeinde und sehr vielen portierten Anwendungen

OpenBSD Maskottchen Puffy/Blowfish



OpenBSD - *“Free, Functional and Secure.”*

- läuft vorzugsweise auf Servern insbesondere Firewalls
- Schwerpunkte: (proaktive) Sicherheit, Standardisierung, Portabilität und starker Kryptographie
- konservatives System welches auf aktuell 16 Plattformen läuft
- Code-Qualität: sauber, einfach, übersichtlich und sicher (zum Beispiel einfach mal einen Blick auf DHCP-Daemon oder beliebigen Kernel-Treiber werfen und mit Linux vergleichen)
- wird oft als das “sicherste Betriebssystem der Welt” bezeichnet

“Only one remote hole in the default install, in more than 8 years!”

Der Fork - das dunkle Kapitel

- OpenBSD ist am 18. Oktober 1995 aus NetBSD herraus entstanden, gegründet durch Theo de Raadt
- deswegen ist die Code-Basis der beiden Projekte teilweise heute noch sehr ähnlich
- damals gab es schwere Differenzen zwischen Theo de Raadt und dem NetBSD Core-Team
- man hatte ihn aus dem Core-Team rausgeworfen und trotz mehrerer Versuche ist man nicht mehr auf einen Nenner gekommen
- insgesamt ein sehr dunkles Kapitel mit vielen Flames und persönlichen Attacken
- Theo de Raadt hat den **kompletten** e-Mail Verkehr der zu dem Fork führte auf seiner Homepage veröffentlicht

Theo de Raadt (* 19. Mai 1968 in Pretoria, Südafrika)

- lebt in Calgary, Kanada wo er an der University of Calgary Informatik studierte
- war 1993 einer der Gründer des NetBSD Projekts und gründete 1995 OpenBSD
- umstrittene Persönlichkeit - Hauptgrund sind seine oft sehr aggressiven/cholerischen Attacken
- “kämpft” seit Jahren aktiv oft “mit allen Mittel” für wirklich freie Software
- Beispiel: Adaptec Raid-Treiber und Dokumentation
- dafür wurde ihm auf der FOSDEM 2005 der FSF Award 2004 von Richard Stallman verliehen

cat /usr/src/usr.bin/mg/theo.c

“Just a minute ago we were hugging and now you, guys, do not love me anymore”
“That’s the most ridiculous thing I’ve heard in the last two or three minutes!”
“I’m not just doing this for crowd response. I need to be right.”
“I’d put a fan on my bomb.. And blinking lights...”
“you have to stop peeing on your breakfast”
“Buttons are for idiots.”
“#ifdef is for emacs developers.”
“You’re not allowed to have an opinion.”
“Quite frankly, SSE’s alignment requirement is the most utterly retarded idea since eating your own shit.”
“Stop wasting your time reading people’s licenses.”
“Linux is fucking POO, not just bad, bad REALLY REALLY BAD”
“penguins are not much more than chickens that swim.”
“Whiners. They scale really well.”
“in your world, you would have a checklist of 50 fucking workarounds just to make a coffee.”

“You can call it fart if you want to.”

“slackers and whiners...”

- OpenBSD ist ein System von Entwicklern für Entwickler: sie programmieren und entwickeln es primär für sich selbst
- vergleichsweise sehr wenig Man-Power: rund 100 Entwickler (mit Commit-Rechten) weltweit
- letztendlich entscheidet immer Theo de Raadt (ähnlich Linus Torvalds beim Linux Kernel) → Cathedral
- trotzdem sehr offene Entwicklung: gute(**!**) Patches werden selbstverständlich akzeptiert → Bazaar
- OpenBSD ist ein sehr **konservatives** System von IT-Professionals die es in ihrer Freizeit entwickeln
- oft sehr rüder und aggressiver Ton von **Machern** auf der Mailinglisten, um sich **“Whiners”** vom Hals zu halten

Sub-Projekte

- aus OpenBSD herraus (durch seine Entwickler) entstanden und entstehen immer wieder einige sehr bedeutende und innovative Neu-Entwicklungen, welche dann portiert werden auf andere Systeme:
  - OpenSSH - wohl die bekannteste und weit verbreitetste SSH-Implementierung
  - OpenNTP - alternative, einfachere und sichere NTP-Implentierung
  - OpenBGP - freie Implementierung des Routing Protokolls
  - PF - alternativer sehr leistungsfähiger Paket Filter
  - CARP - freie alternative zu Ciscos VRRP
  - Zukunft: OpenCVS(!), Apache-Dilemma $\rightarrow$ OpenHTTP(?)

Immer wieder im Mai und November

- im Gegensatz zu vielen anderen Systemen folgt OpenBSD **festen** Release Zyklen
- halbjährliche Stable-Releases - die auch **wirklich** stable sind (aktuell 3.8)
- Verteilung per Kauf-CD, FTP/HTTP, AFS, RSYNC und natürlich CVS
- 3 Flavors:
  - Release - Inhalt auf CD
  - Stable - Release + Sicherheits-Patches
  - Current - aktueller CVS Source zwischen Releases
- vor Stable Release wird CVS eingefroren (FROZEN)
- einmal im Jahr, meistens nach dem Mai Release findet der Hackathon statt

Patches

- Patches für Stable gibt es nur für 2 Releases zurück, also sollte/muss man nach spätestens einem Jahr upgraden
- damit entfällt auch die beliebte Diskussion über Server-Uptime - wenn die Uptime mehr als 1 Jahr ist, kann es kein sicheres System mehr sein
- Patches werden auf Source Basis rausgegeben (als Diffs) $\rightarrow$ bei einem neuen Patch das System neu kompiliert werden $\rightarrow$ kein Hindernis
- eigene Erfahrung: BSD-Kernel auf Athlon 1200 MHZ mit 512MB RAM in weniger als 10 Minuten Linux-Kernel mit Standard-Konfiguration rund 90 Minuten

(pro-aktive) Sicherheit als wichtigste Eigenschaft

- GCC-Erweiterung: gegen Pufferüberläufe (IBMs Stack Smashing Protector, ehemals ProPolice)
- eigenes Speicherschutzkonzept (W^X, für Writeable xor eXecutable),
- geschützte malloc-Implementation auf Basis von mmap (Zufallsadressen, keine Speicherüberlappung, kein korrupter Speicher sondern SIGSEGV und Programmabsturz bei Fehlzugriff)
- Implementation von strlcat und strlcpy (Alternativen zu strncat und strncpy, geschützt gegen Pufferüberläufe)
- zufällige PID-Vergabe
- alle Server-Prozesse per Default als unpriviligierte Benutzer im chroot
- u.v.a. ...

Rund 3000 vorhandene Ports

- **nicht** Bestandteil des Base-Systems, also auch nicht mehr so sicher
- im Vergleich zu FreeBSD und NetBSD sehr wenige Ports, viele grosse wichtige für Desktops (zum Beispiel OpenOffice) fehlen
- sehr konservativ dadurch stabil und mit Augenmerk auf Sicherheit
- wenig Man-Power vorhanden deswegen leider magere Dokumentation, im Gegensatz zum Base-System

Ports → Packages

- Ports sind die Grundlage für Packages
- ein Port besteht in der Regel aus Makefile, Patches (Diffs), MD5-Summen, und Beschreibung
- Port Infrastruktur erzeugt Packages (.tgz - ähnlich RPM oder DEBs) welche installiert/verwaltet werden durch Package-Tools (pkg_add, pkg_delete, pkg_info, pkg_*...) - ähnlich z.B. apt-get
- die komplette Ports-Entwicklung findet auf Mailingliste ports@openbsd.org statt, kein Bug-Tracker oder Request for Packages oder ähnliches...
- wichtig ist es den Ports-Tree und das Base-System in “sync” zu halten
- Ports-Entwicklung findet ausschlieslich auf Basis von Current statt

KedPM

- aus Eigennutz portiert
- pure python port
- Abhängigkeiten von Python Modulen: py-gtk2 und py-crypto

KedPM wurde in Current Ports-Tree (CVS) übernommen und wird mit dem nächsten Stable Release OpenBSD 3.9 Anfang Mai 2006 als Package für alle unterstützten Plattformen mit ausgeliefert.

#cat /usr/ports/security/kedpm/Makefile

```
1  # $OpenBSD: Makefile,v 1.1.1.1 2005/12/04 17:36:51 alek Exp $
2
3  COMMENT=          "application for managing passwords"
4
5  DISTNAME=         kedpm-0.4.0
6  CATEGORIES=       security
7
8  HOMEPAGE=         http://kedpm.sourceforge.net/
9
10 MAINTAINER=       Joerg Zinke <umaxx@oleco.net>
11
12 # GPL
13 PERMIT_PACKAGE_CDROM=    Yes
14 PERMIT_PACKAGE_FTP=      Yes
15 PERMIT_DISTFILES_CDROM=  Yes
16 PERMIT_DISTFILES_FTP=    Yes
17
18 MASTER_SITES=    ${MASTER_SITE_SOURCEFORGE:=kedpm/}
19
20 RUN_DEPENDS=     ${MODPY_EXPAT_DEPENDS} \
21                  :py-gtk2-*:x11/py-gtk2 \
22                  :py-crypto-*:security/py-crypto
23 REGRESS_DEPENDS=${RUN_DEPENDS}
24
25 MODULES=          lang/python
26
27 do-regress:
28         @cd ${WRKSRC} && ${MODPY_BIN} ./run_tests
29
30 .include <bsd.port.mk>
```

KedPM in action...

weiterführende Literatur:


[Ar, 2003] Jacek Artymiak, 2003
Building Firewalls with OpenBSD and PF
Second Edition http://www.devguide.net/


[Lu, 2003] Michael W. Lucas, 2003
Absolute OpenBSD - UNIX for the Practical Paranoid
No Starch Press Verlag, San Francisco.


[OP, 2006] Webseite OpenBSD, 2006
OpenBSD.org
http://www.openbsd.org/


[PN, 2004] Brandon Palmer, Jose Nazario, 2004
Seure Architectures with OpenBSD
Addison-Wesley Verlag.

Vielen Dank für die Aufmerksamkeit!

Fragen?