Das OpenBSD Projekt

Alexander von Gernler

<grunk@steelix.kd85.com>

20. Chaos Communication Congress

Berlin, 27.-29. Dezember 2003

Über diesen Vortrag

Dieser Vortrag ist...

- Eine Vorstellung des OpenBSD-*Projektes*
- Eine Vorstellung des OpenBSD Betriebssystems
- Ein (teilweise subjektiver) Erfahrungsbericht

Er ist nicht...

- Ein Installations-HOWTO für OpenBSD
- Ein heiliger Kreuzzug

Der Autor...

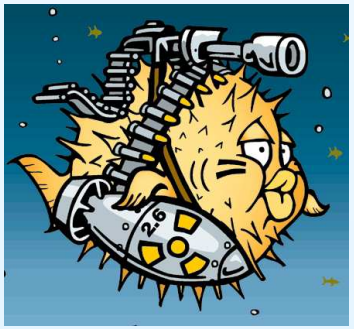
- arbeitet selbst im OpenBSD Translation Project mit
- betreibt den OpenBSD Mirror an der Uni Erlangen

Geschichte der BSD Unices

- 1969 Ur-Entwicklung von UNIX an den Bell Labs durch KEN THOMPSON und DENNIS M. RITCHIE
- ab 1974 Weitergabe der Sourcen an Universitäten, darunter auch die UC Berkeley (**B**erkeley **S**oftware **D**istribution)
- Anfang der 90er Jahre Rechtsstreit mit Bell/AT&T: Entwicklung von BSD/386 ohne strittigen Code
- 1993 Release von 386BSD, später FreeBSD und NetBSD
- 1996 Streit zwischen NetBSD-Team und einem seiner Mitglieder
- THEO DE RAADT forkt nach Ausschluß aus dem NetBSD-Team den Code und ruft das OpenBSD Projekt ins Leben.

Einordnung des Projekts

- **FreeBSD**
  - http://www.FreeBSD.org
  - Stabilität, Performance, Vielzahl von Anwendungen
- **NetBSD**
  - http://www.NetBSD.org
  - Portabilität – läuft auf über 53 Plattformen
- **OpenBSD**
  - http://www.OpenBSD.org
  - Sicherheit, Kryptographie, korrekte Implementation



⇒ **Unzählige Linux-Distributionen, aber nur drei BSDs**

Philosophie von OpenBSD

- So frei wie möglich sein: BSD-Lizenz erlaubt *jedem*, den Quellcode für *jeden* Zweck zu verwenden
- Oberste Priorität auf Sicherheit: Auf Sicherheitsprobleme achten und sie beheben, bevor irgendjemand anders es tut
- Integration starker Kryptographie (IPsec, Kerberos, AFS, Verschlüsselungs-Algorithmen uvm.) mit der Möglichkeit, sie überallhin zu exportieren (Projektsitz in Kanada, keine Exportbeschränkungen)
- Die beste Entwicklungsplattform überhaupt bereitstellen – Zugriff auf den OpenBSD-CVS Baum inklusive
- Standards verfolgen und korrekt implementieren (POSIX, ANSI, X/OPEN etc.)
- Politikfrei bleiben – technischer Fortschritt zählt

Strukturen im Projekt

- THEO DE RAADT als Projektleiter (Coordinator) und *freundlicher Diktator*
- Team der Entwickler mit CVS-Schreibzugriff auf `cvs.openbsd.org` (Committers)
- Jeder, der vernünftigen Code einreicht, kann beitragen (Contributors)

- Team der Übersetzer mit Zugriff auf `steelix.kd85.com`
- WIM VANDEPUTTE für Versand in ganz Europa – `http://www.kd85.com`
- Jährliches Treffen der Entwickler zum *Hackathon* in Calgary
- Finanzierung durch Spenden und Verkauf von Merchandise, Unabhängigkeit als sehr wichtiges Ziel

Das OpenBSD Betriebssystem – Eigenschaften

- Freies Open Source Unix, basierend auf den 4.4BSD Quellen
- Eigenes Binärformat, Emulation für Linux, FreeBSD, Solaris (SVR4), BSD/OS, SunOS und HP/UX
- Läuft auf folgenden Plattformen: i386, Macintosh PPC, Sparc, Sparc64, Alpha, HP 9000 300/400 (hp300) HP 9000 700 (hppa), Mac 68k, Pegasos, Vax
- Utilities und Dämonen für IPsec, VPN-Lösungen, Firewalling, IPv4/IPv6
- Minimalinstallation ca. 100 MB
- Software installierbar als fertige Binärpakete oder aus der Ports-Collection (z. Z. mehr als 2453 Ports verfügbar)

OpenBSD für Linux-User – Verfügbarkeit

- Auslieferung auf 3 CDs jedes halbe Jahr in neuer Release
  - Schickes Artwork mit Aufklebern
  - Neuer OpenBSD Release Song
  - Kostenpunkt ca. EUR 45,–
  - Erhältlich bei Lehmann's oder `kd85.com`

- Mit einer Diskette oder CD (cd34.iso) und Internetanschluss auch kostenlos aus dem Netz installierbar
- Großes Netz an weltweiten Mirrors verfügbar (z. B. `openbsd.informatik.uni-erlangen.de`)
- Bereitstellung von Security-Patches auf der Projekt-Website unmittelbar nach Erkennen und Beheben eines Fehlers
- Tagesaktuelle Mirrors

OpenBSD für Linux-User – Gemeinsamkeiten

> *„BSD = Linux with a twist“*
>
> — CHRISTIAN „naddy“ WEISGERBER

- Oberflächlich: PC einschalten, Bootloader, Kernel bootet, Textkonsole mit Login, X11 möglich
- Alle bekannten Standardbefehle vorhanden (`ps(1)`, `top(1)`, `ifconfig(8)`, `ping(8)`, `vi(1)`, `...`)
- Alle wichtigen Server-Dämonen vorhanden oder nachinstallierbar (`httpd(8)`, `ftpd(8)`, `named(8)`, `sendmail`, `sshd(8)`, `rsyncd`, `cvsupd`, `nptd`, `...`)
- Gängige Softwarepakete als Ports vorhanden (MySQL, Postgres, nmap, Gimp, xmms, MPlayer, dia, Mozilla, gkrellm, . . .)
- Breite Hardwareunterstützung, auch was USB Geräte und Soundkarten betrifft

OpenBSD für Linux-User – Unterschiede

- BSD **F**ast **F**ile**s**ystem `ffs` – vergleichbar mit Linux `ext2`
- Erweiterung: *SoftUpdates*, vergleichbar mit einer Journaling-Funktion wie bei `ext3` vorhanden
- Linux ist ein SystemV Unix, OpenBSD ist ein BSD
- Kernel und Userland bilden feste Einheit, können nur *gemeinsam* upgedatet werden. $\Rightarrow$ passen immer zusammen

OpenBSD für Linux-User – Unterschiede

- BSD **F**ast **F**ile**s**ystem `ffs` – vergleichbar mit Linux `ext2`
- Erweiterung: *SoftUpdates*, vergleichbar mit einer Journaling-Funktion wie bei `ext3` vorhanden
- Linux ist ein SystemV Unix, OpenBSD ist ein BSD
- Kernel und Userland bilden feste Einheit, können nur *gemeinsam* upgedatet werden. $\Rightarrow$ passen immer zusammen
- Keine Probleme wie:
  - `lvmtools` passen nicht zum Kernel-LVM
  - `iptables` will eine andere Version von `netfilter` im Kern
  - `reiserfsck` hat doch mit dem alten Kern noch getan?!

OpenBSD für Linux-User – noch mehr Unterschiede

- Installation von Softwarepaketen meist nicht als Binary (`.rpm, .deb`), sondern durch die Ports-Collection
  - Sammlung von Makefiles unter `/usr/ports`
  - Tagesaktuell per CVS updatebar
  - Kümmert sich um das Holen, Entpacken, Checken, Compilieren, Installieren
  - `root@vario:/usr/ports/editors/vim/stable% make install`
  - Ports Collection berücksichtigt Abhängigkeiten zwischen den Paketen (Dependencies)
  - Danach ist alles erledigt
- Aber auch Bauen von Binärpaketen möglich
- Installation und Verwaltung mit `pkg_add(1), pkg_info(1), pkg_create(1), pkg_delete(1)`

OpenBSD ist einfach

- nur ein Runlevel, Reboot, Single-User und Halt
- exzellente Manpages mit guten Beispielen
- Der User ist am Anfang nicht aufgeschmissen
  - `grunk@vario:~% man afterboot`
  - `grunk@vario:~% man intro`
- Zwei Versionen für den Anwender per CVS auscheckbar
  - `-stable` ist die Release-Version zusammen mit aktuellen Sicherheitspatches. Geeignet für den Produktionsbetrieb (der OpenBSD Mirror in Erlangen benutzt `-stable`
  - `-current` ist die „cutting edge“ der Entwicklung und (aus eigener Erfahrung) nicht minder stabil: Der Autor benutzt `-current` auf seinen Rechnern daheim

Sicherheit

- Swapspace verschlüsselbar

```
root@vario:~% sysctl -w vm.swapencrypt.enable=1
```

- `/dev/random` nutzt viele verschiedene Entropiequellen für guten Zufall:
  - Mausinterrupts, Netzwerk-I/O, Tastaturanschläge, Disk-I/O, Hardware-RNG (wenn vorhanden)
- Entropie wird verwendet, um Systemverhalten für Angreifer unvorhersagbar zu machen:
  - Prozess-IDs, IDs von UDP-Paketen, Inode-Nummern, temporäre Dateinamen in `mktemp(3)`, Salts für Passwort-Algorithmen, TCP-Sequenznummern und viel mehr
- ProPolice Stack Protection erschwert Angriffe mit Buffer Overflows

Sicherheit – still more

- `W^X` (Write XOR Execute) für Speicherseiten verhindert weitere Gemeinheiten
- Firewalling unter OpenBSD mit `pf(4)`
  - regelbasierter Paketfilter
  - kann NAT verschleiern (`scrub reassemble tcp`, siehe `pf.conf(5)`)
  - kann aufgrund von Remote-OS Entscheidungen treffen (passive OS Fingerprinting)
  - beherrscht Alternative Queuing (`altq(9)`)
  - kann Zustände mit redundanten Peers synchronisieren
- minimalistisches Default-Install sorgt für schlankes, sicheres Standardsystem
- tagesaktueller `-stable` Branch hat bewährte Features der Release, zusammen mit aktuellen Security-Patches

Sicherheit – Securelevels (`securelevel(7)`)

- `kern.securelevel=1`
  - Securelevel kann nicht mehr gesenkt werden
  - `/dev/mem, /dev/kmem` nicht mehr schreibbar
  - Partitionen gemounteter Dateisysteme können nicht direkt geschrieben werden
  - Dateiflags wie `immutable` oder `append-only` greifen
  - Kernelmodule können weder geladen noch entfernt werden
- `kern.securelevel=2`
  - Alles von Securelevel 1 gilt
  - Raw Disks dürfen überhaupt nicht geschrieben werden
  - Die Systemzeit kann nicht mehr zurückgesetzt werden
  - Firewallregeln können nicht mehr verändert werden
  - Der Kerneldebugger kann nicht mehr aktiviert werden

Interessante Features

- `spamd(8)`: Teergrube für Verbindungen von Rechnern, die als Spamversender bekannt sind
- `authpf(8)`: Erlaubt Benutzern mit Rechnern im lokalen Netz das Passieren der Firewall, wenn Authentifikation per `ssh` erfolgt
- `systrace(1)`: Aufstellen und Einhalten von Privilegienprofilen bei Syscalls für einzelne Programme
- Unterstützung von Kryptohardware (Kryptobeschleuniger), so dass keine Konfiguration mehr nötig ist – einstecken, läuft
- Die meisten Dienste unter OpenBSD laufen entweder in einer `chroot(8)`-Umgebung (`httpd(8), ftpd(8)`) oder mit spezieller *Privilege Separation* (X11, `syslogd(8)`) und nur ganz wenige mit setuid/setgid `root`

Syscall Kontrolle mit `systrace(1)`

- Verbietet oder erlaubt einzelnen Programmen explizit, Syscalls auszuführen oder nicht
- Weiteres Einschränken von bereits im `chroot` laufenden Dämonen möglich („Hä, wofür will mein Apache eine root-Shell spawnen!?“)
- Auch *privilege elevation* möglich: User-Prozesse können einzelne Syscalls z. B. mit `root`-Rechten ausführen, ohne setuid-Bit zu haben
- Entscheidungen basierend auf Policy-Dateien in `${HOME}/.systrace`
- Matches auf Gleichheit, Globbing, RegEx, Substrings möglich. Überprüfbar u. a. sockaddr, filename, gid
- Erstellen der Profile per Hand, von Vorlagen aus dem Netz oder durch eigene Generierung

Noch mehr Syscall Kontrolle mit `systrace(1)`

- Profile generieren:

```
root@vario:~% systrace -A /usr/libexec/ftpd
root@vario:~% ps ax | grep ftpd
24421 ?? Ixs    0:00.00 /usr/libexec/ftpd
12929 ?? Is     0:00.01 systrace -A /usr/libexec/ftpd
root@vario:~%
```

- Profile durchsetzen:

```
root@vario:~% systrace -a /usr/libexec/ftpd
```

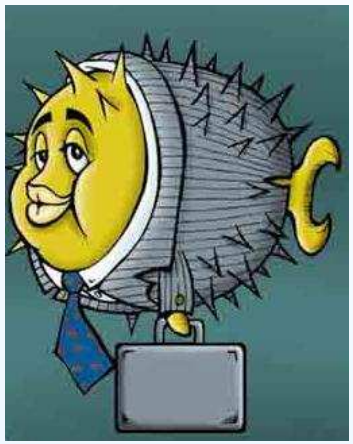
- Profile interaktiv verändern: An einen laufenden `systrace` attachen

```
root@vario:~% systrace -p 12929 /usr/libexec/ftpd
```

  Bei einem nicht authorisierten Aufruf poppt ein kleines GUI auf, das eine Entscheidung verlangt

OpenBSD ist interessant für Firmen

Bedeutende Referenz-Anwender



- Universitäten und NGOs (z. B. University of Alberta (Canada), University of Minnesota)
- kommerzielle Firmen (z. B. Adobe, CORE SDI, Alteon Networks)
- große Internetprovider (z. B. Calyx, Anonix, Globalwire Communications)
- behauptet: Einsatz bei Militär und Geheimdiensten (nicht nachprüfbar)

„Free, functional and secure“

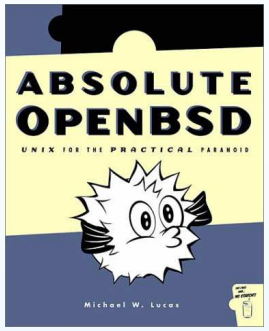
- kostenlos, sicher, portabel, standardkonform
- BSD-Lizenz erlaubt kommerzielle Verwertung auch als Closed Source

Schwachpunkte von OpenBSD

- (Noch) nicht Multiprozessorfähig
- Performance bleibt hinter der von FreeBSD und Linux zurück
- Ports-Tree im Vergleich zu dem von FreeBSD noch klein
- Keine automatische Prüfung des Ports-Trees auf Aktualität und Sicherheit mit Standardtools
- Händische Updates von Version zu Version erfordern Sachverstand
- Noch kein bzw. unzureichendes Energiemanagement für portable Geräte, ACPI noch überhaupt nicht supported
- Filesystemchecks von gecrashten `ffs` Partitionen dauern noch so lange wie bei `ext2`, obwohl SoftUpdates einen `fsck(8)` im Hintergrund ermöglichen würden

Literatur

Absolute OpenBSD

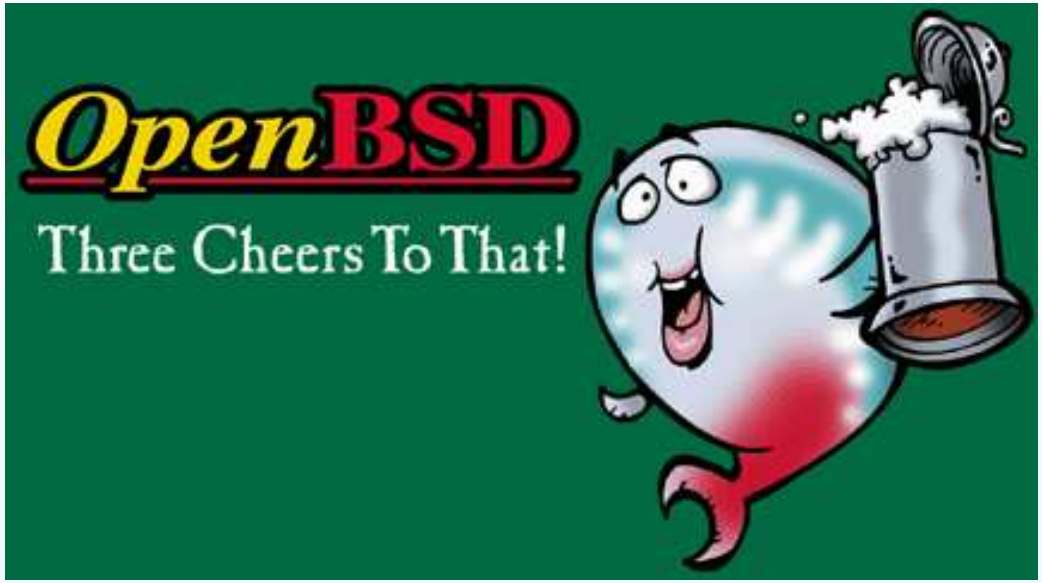


- von MICHAEL W. LUCAS, Juli 2003
- ISBN 1-886411-99-9, 489 Seiten, ca. EUR 40,-
- erhältlich im Buchhandel, `www.kd85.com` oder am OpenBSD Stand

Building Firewalls with OpenBSD and PF

- von JACEK ARTYMIAK, Juli 2003
- ISBN 83-916651-4-3, 248 Seiten, ca. EUR 40,-
- erhältlich im Buchhandel, `www.kd85.com` oder am OpenBSD Stand

Noch Fragen?



- Folien erstellt mit LaTeX, `prosper`, `make(1)` und `cvs(1)` unter OpenBSD 3.4 / i386
- Quellcode der Folien auf Anfrage: `<grunk@steelix.kd85.com>`
- Mails: 72 Zeichen pro Zeile, keine HTML-Mails