Es muss nicht immer Linux sein: Die Vorzüge von OpenBSD

Hans-Jörg Höxer
<Hans-Joerg.Hoexer@yerbouti.franken.de>

24. November 2002

Disclaimer

no flaming :-)

Überblick

- Geschichte der BSD-Unixe
- Ziele des OpenBSD-Projekts
- Unterschiede zu Linux
- Besonderheiten von OpenBSD
- OpenBSD in der Praxis

Geschichte

- Ken Thompson & Co. entwickelten UNIX (1969) bei Ball Labs
- ab 1974 Sourcen wurden an Universitäten weiter gegeben, unter anderem auch UC Berkley $\rightarrow$ Berkley Software Distribution BSD
- Anfang 90er: Rechtsstreit mit Bell/AT&T $\rightarrow$ BSD/386 bzw. BSD4.4-lite von Berkley Software Design, Inc. (BSDI)
- 1993 Release von 386BSD, später FreeBSD bzw. NetBSD

OpenBSD vs. NetBSD

- 1996 zerstritt sich Theo de Raadt (arbeitet an NetBSD für Sparc) mit dem NetBSD-Team – und wurde “gefeuert”
- gründete sein eigenes Projekt “OpenBSD”
- als Ausgangspunkt dienten die NetBSD-Sourcen

Einige Ziele des OpenBSD-Projekts

- bestmögliche Entwicklungsplattform – vollständiger Sourcecode über CVS verfügbar
- Verwendung von Sourcecode mit brauchbarere Linzenz – BSD oder GPL, keine NDA, etc.
- besonderes Augenmerk auf Sicherheitsprobleme
- möglichst maschinen-unabhängige Sourcen – Unterstützung vieler verschiedener Hardwareplattformen

Security

- “Try to be the #1 most secure operating system”
- vollständiges Offenlegen von Sicherheitsproblemen
- systematisches Auditing des Sourcecodes – Sicherheitslöcher vs. Bugs
- “Secure by Default” – sinnvolle und vorsichtige Default-Konfiguration
- Kryptographische Methoden zur Lösung von Sicherheitsproblemen (IPsec, OpenSSH, Blowfish für Passwörter, etc.)

Unterschiede zu Linux

OpenBSD ist ein “Komplett-System”

- Kernel + User-Land
- keine unterschiedlichen Distributionen
- Sourcecode steht vollständig zur Verfügung – “self-hosting”
- Release alle 6 Monate (Mai und Dezember) als CD, dazwischen Snapshots über FTP

Unterschiede zu Linux

Release besteht aus tar-Archiven

- Basissystem, `/etc`, Compiler und Entwicklungs-Utilities, Manpages, BSD-Spiele (`fortune`, `trek`, `wargames`, ...), zusätzliche Dokumentation (`/usr/share`)
- komplettes XFree86 Version 4
- Sourcen (Kernel + User-Land, XF4 extra)
- Zusatzsoftware als “Packages”

Software aus dem Basissystem

- `ksh` und `csh`, `vi` und `mg` als Texteditoren
- die “üblichen” Utilities – `top`, `ps`, `ifconfig`, `ping`, `lynx`, etc.
- `httpd` (Apache/1.3.27), `named` (4.9.11), `ftpd`, `sendmail` (8.12.6), OpenSSH
- Utilities und Dämonen für IPsec, VPN, Firewalling, IPv4/IPv6, NFS, ...

→ vollständiges Serversystem mit ca. 80Mb

Erweitertes Grundsystem

- XFree86 mit `fvwm`, `xterm`, etc.
- GNU-Entwicklertools: `gcc`, `gdb`, `binutils`
- Packages: `vim`, LaTeX, GnuPG, `mutt`

→ typische Installation (Workstation am Internet) umfasst ca. 400Mb, mit Sourcen ca. 1000Mb

Besonderheiten von OpenBSD – Kryptographie 1

Kryptographie kann die Sicherheit eines Systems verbessern:

- KerberosIV und KerberosV (verteilte Authentifizierung, z. B. für X11)
- OpenSSH – sichere Remote-Logins
- IPsec-Stack seit 1997 – VPNs, WLAN, etc.
- Pseudo-Zufallszahlen (PIDs, IDs für IP, ISS für TCP,...) – Entropie-Pool

Besonderheiten von OpenBSD – Kryptographie 2

Chiffren (3DES, Blowfish, CAST,...) und Hash-Funktionen für Kernel und Userland:

- `libc`: Blowfish und MD5 für Passwörter
- IPsec: ESP/AH, ISAKMP/IKE, Photuris
- `libssl` – SSL als de-facto-Standard
- Verschlüsseln des Swap-Spaces

Besonderheiten von OpenBSD – Kryptographie 3

OpenBSD unterstützt seit April 2000 Hardware-Krypto-Beschleuniger:

- z. B. Hifn 7751 und 7951 (Soekris VPN1201)
- ca. 64Mbit/s mit 7951
- “kleine” CPUs (486, 586) sind leistungsfähig genug für VPN-Gateways, wenn sie durch Krypto-Beschleuniger unterstützt werden

Besonderheiten von OpenBSD – Advisories

- Errata-Seiten für jeden Release
- Source-Patches über FTP oder CVS
- mit den Sourcen auf CD und Kontrolle der Patches Schutz vor “Trojanisierung”, etc. (siehe `sendmail` oder `tcpdump`)

Besonderheiten von OpenBSD – Man-Pages

- sorgfältig geschrieben
- kompakt
- mit sehr guten Beispielen versehen – ideale Grundkonfigurationen für eigene Experimente
- Dokumentation der Kernel-Interna (Kapitel 9)

→ “Die Man-Pages sind hervorragend!”

OpenBSD in der Praxis – `systrace`

- Überwachung von Systemaufrufen
- Kontrolle von Systemaufrufen über “Policies”
- Sandboxing für nicht vertrauenswürdige Applikationen
- zusätzlicher Schutz vor Bugs für Dämonen, etc.
- Policies können interaktiv erstellt werden

OpenBSD in der Praxis – `authpf` 1

- User-Shell für Authentifizierungs-Gateways
- User authentifiziert sich über SSH an Gateway
- `pf(4)` Regeln werden für die aktuelle IP-Adresse des Users individuell angepasst
- `authpf`-Session werden über `syslog(8)` protokolliert

OpenBSD in der Praxis – authpf 2

```
...
# rdr ftp for proxying by ftp-proxy(8)
rdr on $internal_if proto tcp from $user_ip to any port 21 \
    -> 127.0.0.1 port 8081

# allow out ftp, ssh, www and https only, and allow user to negotiate
# ipsec with the ipsec server.
pass in quick on $internal_if proto tcp from $user_ip to any \
    port { 21, 22, 80, 443 }
pass in quick proto udp from $user_ip to $ipsec_gw port = isakmp \
    keep state
pass in quick proto esp from $user_ip to $ipsec_gw
...
```

Fragen?

http://www.openbsd.org/