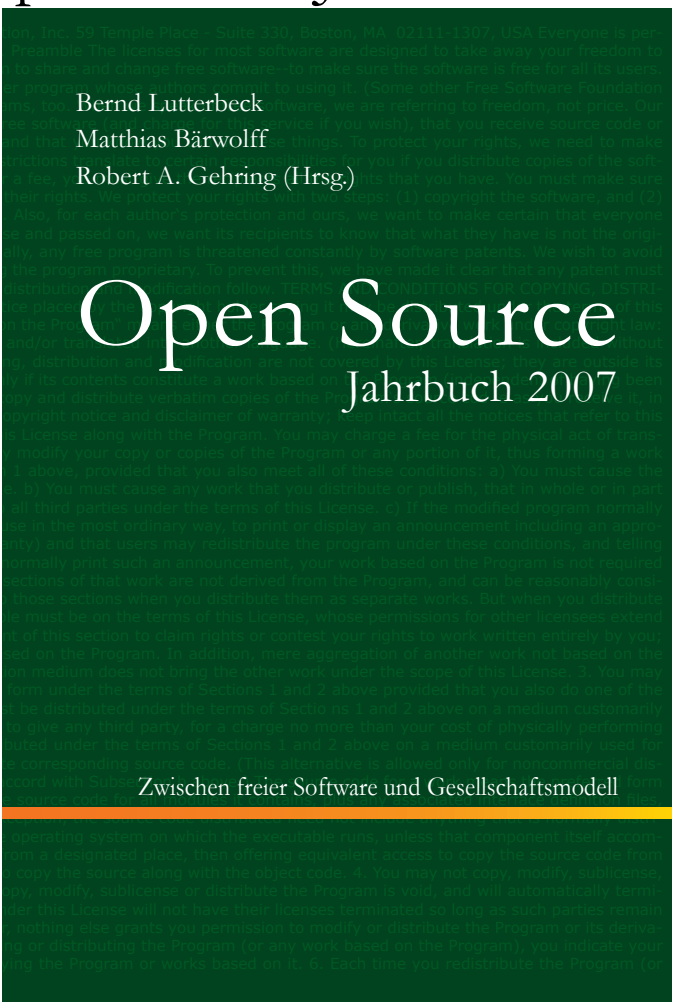
Dieser Artikel ist Teil des

Open Source Jahrbuchs 2007



erhältlich unter www.opensourcejahrbuch.de.

Die komplette Ausgabe enthält viele weitere interessante Artikel. Sie können diesen und andere Artikel im Open-Source-Jahrbuch-Portal kommentieren oder bewerten: www.opensourcejahrbuch.de/portal/. Lob und Kritik sowie weitere Anregungen können Sie uns auch per E-Mail mitteilen.

NetBSD – Das portabelste Betriebssystem der Welt*

HUBERT FEYRER, STEFAN SCHUMACHER UND MARK WEINEM

NetBSD ist ein modernes, vollständig freies Betriebssystem, das kontinuierlich weiterentwickelt wird. Es wird wie viele interessante Open-Source-Projekte leider wenig beachtet. Dabei eignet sich das NetBSD-Betriebssystem für jeden Anwendungszweck und es gibt kaum einen Computer, auf dem es nicht genutzt werden kann. Wegen seiner Qualität und liberalen Lizenz wird es ebenso gerne in der Lehre wie bei der Entwicklung neuer kommerzieller Produkte eingesetzt. Dieser Artikel beschreibt die wichtigsten Merkmale von NetBSD sowie dessen Einsatzmöglichkeiten im privaten und im kommerziellen Umfeld. Eine Betrachtung des pkgsrc-Paketsystems, das auch unabhängig von NetBSD eingesetzt werden kann, schließt den Artikel ab.

Schlüsselwörter: Betriebssystem · BSD · Distribution · NetBSD

1 Was ist NetBSD?

NetBSD[1] wurde 1993 als Nachfolgeprojekt des von der *University of California, Berkeley* herausgegebenen UNIX-ähnlichen Betriebssystems *Berkeley System Distribution (BSD)* gestartet. Es ist ein vollständiges, UNIX-ähnliches Betriebssystem, das den aktuellen Open-Source- und Security-Kriterien entspricht. Gleichzeitig werden Industriestandard-APIs und Vernetzungsprotokolle sowie eine große Anzahl an Hardware-Plattformen unterstützt. NetBSD eignet sich dabei für alle Anwendungsgebiete: robuste Server-Systeme und Workstations ebenso wie Handheld-Rechner und Embedded-Systeme.

Seit seiner Entstehung ist NetBSD an der Spitze der Entwicklung von Open–Source-Betriebssystemen anzutreffen. In vielen Fällen diente NetBSD als vollständige Grundlage oder Referenz für andere Projekte.

* Enthält Text aus: NetBSD „allgemeiner“ und Security Flyer, von Stefan Schumacher, 08. Jan. 2006; sowie clt6-pkgsrc.tex von Hubert Feyrer.

1 Siehe auch http://www.netbsd.org.

NetBSD wird oft gewählt, um neu entwickelte Hardware anzusteuern, u. a. in den Bereichen Netzwerk-Computer, Ein-Platinen-Rechner und selbst Webcams und Roboter. Weiterhin wird es weltweit im Bereich der Netzwerkentwicklung eingesetzt. Internet-Service-Provider benutzen NetBSD wegen seines breiten Spektrums an Netzwerkmöglichkeiten, und begeisterte Liebhaber entscheiden sich für NetBSD aufgrund seiner breiten Hardware-Unterstützung.

Das System läuft auf einer großen Palette von Hardware-Plattformen, angefangen von der *VAX 11/750* bis zu modernsten PCs und Windows-CE-Palmtops. Egal, ob man den alten Computer vom Dachboden entstauben oder einen neuen Opteron-Server in Betrieb nehmen will: NetBSD ist bereit dafür. Durch das einfache Entfernen optionaler Komponenten eignet sich NetBSD auch sehr gut für Embedded-Systeme. Zurzeit unterstützt NetBSD 55 verschiedene Hardware-Plattformen auf insgesamt 17 CPU-Architekturen von der *DEC Alpha* über *Sparc64* bis hin zum *ARM* – alle aus demselben Quellcode.

Die am NetBSD-Projekt beteiligten Entwickler legen hohen Wert auf sauberen Quellcode, ausgereifte Lösungen, Sicherheit und Portabilität. Aufgrund dieser Philosophie bietet NetBSD ausgezeichnet les- und wartbaren Code, der die Grundlage für viele weitere Projekte bildet. Um die legendäre Portabilität des Systems auf mehr als fünfzig verschiedene Plattformen zu gewährleisten, wird strikt zwischen maschinenabhängigem und maschinenunabhängigem Quellcode getrennt. Dies führt neben der bekannten und hochgeschätzten Stabilität auch zu einem exzellent strukturierten Systemkern (Kernel), der in einer einzigen Konfigurationsdatei angepasst werden kann. Weiterhin hält sich das Projekt an bestehende Dokumentationen und Standards, wie beispielsweise *POSIX*[2], und präsentiert sich so als ein hervorragendes System zur Forschung und Lehre im akademischen Umfeld.

Das Cross-Kompilieren von Kernel, *Userland* und dem *X Window System* als grafischem Unterbau wird von den Standardwerkzeugen unterstützt. Damit kann man das komplette System für ältere, langsamere aber auch Industrie- und Embedded-Rechner auf einem schnelleren Rechner einer beliebigen Architektur kompilieren.

NetBSD bietet ein überschaubares System, in dem Kernel und *Userland* integriert sind. Weitere Software ist durch das NetBSD-Paketsystem einfach installierbar. Durch die Trennung von Anwendungen und Basissystem können Updates von Anwendungen nichts am Basissystem ändern. Der vollständige Quellcode ist, inklusive Entwicklungsgeschichte, via anonymem *CVS*[3], *CVSweb*[4], *SUP*[5] und *rsync*[6] verfügbar.

2 Aus dem IEEE Standard 1003.1-1988 herausgehend wird hier definiert, wie sich ein UNIX(-ähnliches)-System verhalten soll. Der Standard wird in der aktuellsten Version von der *Open Group* als wesentlicher Bestandteil der *Single UNIX Specification* herausgegeben und ist frei erhältlich. Siehe dazu auch http://www.opengroup.org/austin/papers/posix_faq.html.

3 Checkout via anoncvs@anoncvs.netbsd.org:/cvsroot.

4 Siehe http://cvsweb.netbsd.org.

5 *SUP* steht für *Software Update Protocol*.

6 Siehe http://rsync.samba.org.

Außerdem erstellt das NetBSD-Projekt zu jedem neuen Release Images für Installations-CDs, die von weltweit verteilten FTP-Spiegeln[7] bezogen werden können. Mehrere Anbieter vertreiben NetBSD auf CD und DVD.[8]

Die Unterstützung der unterschiedlichsten Netzwerktechnologien wie *WLAN IEEE 802.11*, *Gigabit-* und *10-Gigabit-Ethernet*, *ATM*, *Bluetooth*, *HIPPI*, *FDDI*, *HSSI*, *ARCnet* und *Token-Ring* ist bereits im Basissystem gegeben. NetBSD war zudem das erste Open-Source-Betriebssystem, das *USB*, *USB2* und *PCMCIA Audio* unterstützte.

Zusätzlich zum System bietet NetBSD eine Vielzahl ausgereifter Dokumentationen. Neben einem umfangreichen Benutzerhandbuch, das viele Aspekte der Administration beschreibt, gibt es mehrere Artikel und Bücher, die für Entwickler den Aufbau des Systems im Detail beschreiben. Standardmäßig kommt NetBSD mit einer umfangreichen Sammlung an *man pages* zu allen Programmen, Bibliotheken, Funktionen und sonstigen Themenfeldern des Systems. Weiterhin existieren zahlreiche Informationskanäle und Hilfsquellen im gesamten Internet, von Mailinglisten über Newsgroups bis hin zu Webforen und Wikis.

Auch außerhalb des Internets ist NetBSD auf Messen und Veranstaltungen präsent: Entwickler und Anwender präsentieren das System sowohl in Vorträgen als auch live, Interessierte werden beraten und bei offenen Fragen und Problemen unterstützt.[9]

2 Projektstruktur

BSD unterstützte zum Zeitpunkt seiner Entstehung bereits eine breite Palette an unterschiedlichster Hardware, und so lag das Vorhaben nahe, NetBSD als System zu erstellen, das diese verschiedenen Hardware-Architekturen unterstützt und weiter ausbaut. Hier grenzt sich auch NetBSD vom zweiten BSD-Nachfolger *FreeBSD* ab, dessen Fokus auf Optimierung für die damals an Verbreitung gewinnende PC-Plattform liegt. Zur Namensgebung bei NetBSD hat das damals ebenfalls heranwachsende Internet beigetragen – Grundidee war es, dass sich alle Interessierten über das „Netz" (engl. *net*) koordinieren und zusammenarbeiten.

Die Koordination des Projekts sowie die Integration der Arbeiten aus Forschungseinrichtungen, Universitäten und Firmen wurden vom Core-Team überwacht. Mit der Zeit wuchs die Anzahl der Beiträge, und es wurde klar, dass nicht mehr nur das Core-Team Schreibzugriff auf den NetBSD-Quellcode brauchte, sondern auch weitere Entwickler mit Schreibrechten auf den Quellcode ausgestattet werden mussten. So konnte das Core-Team entlastet und der Entwicklungsprozess vereinfacht werden. Heute arbeiten weltweit mehr als 300 Entwickler für das NetBSD-Projekt.

7 Siehe http://ftp.netbsd.org/pub/NetBSD/.

8 Siehe http://www.netbsd.org/Sites/cdroms.html.

9 Eine Übersicht der Veranstaltungen, auf denen NetBSD präsentiert wird, findet sich unter http://www.netbsd.org/gallery/events.html.

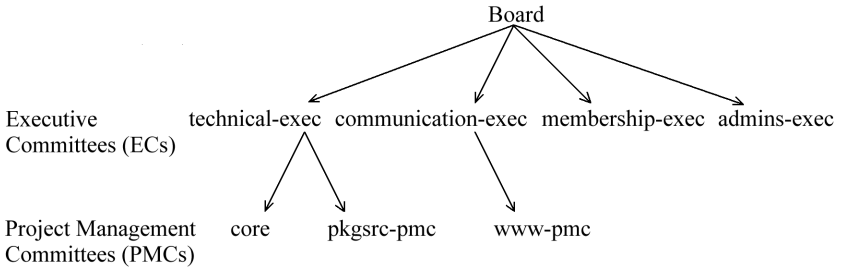


Abbildung 1: Organisation der NetBSD Foundation

Mit der Anzahl der Entwickler wuchs auch der IT-Aufwand: Es reichte nicht mehr aus, dass ein einzelner Rechner für Entwicklung (*CVS*), Bereitstellung (*FTP*), Fehlerdatenbank (*GNATS*) und Mailinglisten zuständig war, und bei der Verteilung der Aufgaben auf mehrere Rechner wollten weitere Aspekte berücksichtigt werden, wie Absicherung gegen unberechtigte Zugriffe, Datensicherheit durch Backup, sowie Wartung des Betriebssystems durch möglichst gleiche Hard- und Software.

Es wurde schnell klar, dass weitere Verwaltungsstrukturen nötig waren, u. a. ein ganzes Team von Administratoren, das die projektinternen Rechner betreut. Weitere Verbindlichkeiten entstanden, oft wurden und werden hier rechtsverbindliche Ansprechpartner benötigt. Dies galt auch für die Annahme von Spenden. Als Projekt von Freiwilligen hat NetBSD keine Firma im Hintergrund, die sich um Rechner, Administration, Anbindung und nicht zuletzt Finanzierung kümmert.

Aus diesem Bedarf an Formalisierung wurde die *NetBSD Foundation* gegründet. Ziel war und ist es, einerseits rechtsverbindlich für das NetBSD-Projekt Verträge für Hardware, Colocation und Netzwerk-Connectivity eingehen zu können, aber auch als Eigentümer der Werte des NetBSD-Projektes auftreten zu können: Hierunter fallen neben der Hardware der diversen Server v. a. das geistige Eigentum wie das Urheberrecht am NetBSD-Betriebssystem und *pkgsrc* sowie die zugehörigen Warenzeichen. Zudem kann die *NetBSD Foundation* in den USA als gemeinnützig anerkannter Verein Spenden annehmen und entsprechende Quittungen ausstellen, was zu einer einfacheren Unterstützung des NetBSD-Projekts durch Firmen führt.

An der Organisationsstruktur der *Internet Engineering Task Force (IETF)* orientiert, geschah dann auch die Aufteilung der *NetBSD Foundation* in mehrere sog. *Executive Committees (ECs)*, welche die Arbeit von mehreren *Project Management Committees (PMCs)* sowie letztendlich der in diesen Gruppen arbeitenden NetBSD–Entwickler überwachen (siehe Abbildung 1). Dies alles geschieht unter der Oberaufsicht des *Board of Directors*, dem Vorstand.

Die technische Entwicklung wird von *technical-exec* überwacht, wobei hier zwei Teilbereiche existieren: Die Entwicklung des NetBSD-Betriebssystems wird wie gehabt vom Core-Team geleitet, über die Entwicklung von *pkgsrc* als systemüber-

greifende Paketsammlung wacht das *pkgsrc-PMC*. Die Öffentlichkeitsarbeit wird von *communication-exec* überwacht, wobei hier die Wartung des Webservers an das *www–PMC* ausgegliedert ist.

Membership-exec ist die Personalverwaltung des NetBSD-Projektes und rekrutiert – üblicherweise in Zusammenarbeit mit weiteren Entwicklern – neue Entwickler und kümmert sich um die nötigen Unterlagen, um die Mitgliedschaft der neuen Entwickler in der *NetBSD Foundation* in Übereinstimmung mit der Satzung[10] zu sichern. Der *finance-exec EC* ist die Abteilung, die sich um den Finanzfluss des NetBSD–Projekts kümmert: Annahme von Spenden, Ausstellung von Spendenquittungen, aber auch Verwaltung der Konten des Projektes sowie finanzielle Abwicklung von Käufen z. B. für neue Rechner oder Ersatz-Hardware, wenn 'mal wieder eine Festplatte kaputt ist oder zu klein wird.

Die letzte, jedoch nicht unwichtigste Gruppe bildet *admins-exec*, das sich um die Betreuung der Rechner des NetBSD-Systems kümmert, Software und Betriebssysteme installiert und aktualisiert, Datensicherung betreibt und auch über die Sicherheit der Rechner wacht.

Neben den eingetragenen Entwicklern des NetBSD-Projekts tritt die zweite wichtige Gruppe in Abbildung 1 nicht explizit auf: Anwender, Administratoren, Programmierer und Entwickler. Obwohl für sie keine Mitgliedschaft in der *NetBSD Foundation* erforderlich ist, tragen sie doch entscheidend zur Weiterentwicklung von NetBSD bei: durch Erfahrungs- und Fehlerberichte, Einsenden von Erweiterungen für neue Features, Treiber oder neue Subsysteme, Dokumentation sowie auch durch ihre Entscheidung für NetBSD beim Entwickeln neuer Produkte und Dienste und nicht zuletzt durch finanzielle Unterstützung.

Wie bei allen Open-Source-Projekten spielt auch bei NetBSD neben der Projektleitung die Community die wichtigste Rolle für die Zukunft des Projektes.

3 NetBSD in der Geschäftswelt

3.1 Die BSD-Lizenz

Die Stärken von Open-Source-Betriebssystemen gegenüber herkömmlichen kommerziellen Betriebssystemen werden von immer mehr Firmen erkannt. Beim Wunsch, Produkte auf Basis der existierenden Open-Source-Systeme anzubieten spielt deren Lizenz eine wichtige Rolle. Während z. B. die *GNU General Public License (GPL)* explizit fordert, dass Änderungen an GPL-Code wieder der Öffentlichkeit zugängig gemacht werden müssen, ist dies bei NetBSD nicht der Fall, da hier die BSD-Lizenz benutzt wird. Diese erlaubt es, NetBSD als Basis für kommerzielle Systeme zu verwenden und eigene Anpassungen zum Schutz von Wettbewerbsvorteilen für sich zu behalten.

10 Siehe http://www.netbsd.org/Foundation/bylaws.html.

3.2 Offenlegung zur Wartung

Obwohl Herstellern damit die Freiheit eingeräumt wird, ihre Änderungen am Code für sich zu behalten, so spricht andererseits doch eine Reihe von Gründen dafür, Änderungen trotzdem wieder an NetBSD zurückzugeben:[11] Zum einen können künftige Entwickler und Projekte auf ihren gemachten Erfahrungen aufbauen. Zum Anderen kann dadurch ein Peer-Review für den Quellcode entstehen, was vor allem bei sicherheitskritischen Bereichen besonders relevant ist. Zuletzt spricht der Wartungsaspekt für die Rückführung von Änderungen: Da NetBSD permanent weiterentwickelt wird, müssen sonst die Hersteller die am NetBSD-Code durchgeführten Änderungen selbst nachvollziehen, um ihre Software zu integrieren. Bei einer Rückführung des Codes an das NetBSD-Projekt werden diese Änderungen im Rahmen der Entwicklungsarbeiten an NetBSD eingepflegt.

3.3 Anwendungsbeispiele

Hier einige Anwendungen, bei denen NetBSD als Basis fungierte:

- Teile des Systems wurden in andere Betriebssysteme integriert, z. B. *Apples Darwin*[12], *Castle Technology Ltds RiscOS*[13], *FMSLabs' RTCore/BSD*[14], *QNX*[15].
- Diverse WLAN-Router von *Allied Telesys*[16], *IIJ/Root Inc*[17].
- *Avocent* KVM Switches[18].
- *Brocade Rhapsody* SAN Switches[19].
- *TeamASAs Npwr*[20]: *NAS* Server mit *iSCSI*, *FiberChannel*, *Xscale*.
- *Seclaritys SiNic*[21]: *Wireless router on a card* für den Militäreinsatz.
- *Ricoh Co. IPSiO*[22]: Kopierer, Drucker, FAX, Scanner.

11 Siehe *Corporate reasons for BSD over GPL* unter http://www.feyrer.de/NetBSD/blog.html/nb_20050209_2138.html.
12 http://www.apple.com/macosx/
13 http://www.drobe.co.uk/riscos/artifact1111.html
14 http://www.fsmlabs.com
15 http://www.qnx.com/licensing/published/eula/TPOSLTG1_01.html
16 http://www.allied-telesis.co.jp/products/list/wireless/wr54id/catalog.html
17 http://www.iij.ad.jp/en/pressrelease/2005/0406.html
18 http://www.avocent.com/web/en.nsf/AttachmentsByTitle/590-356-001D(SVIP).pdf/%24FILE/590-356-001D(SVIP).pdf
19 http://www.byteandswitch.com/document.asp?doc_id=29068
20 http://www.teamasa.com/npwr_netbsd.shtml
21 http://www.seclarity.com/products/wireless/
22 http://www.ricoh.co.jp/imagio/mono/

– Überwachungskameras und Webcams von *SGI*[23], *Panasonic*[24] und *Brains Inc.*[25].
– Zahllose Evaluation-Boards, SBCs und Spezialcomputer von *Broadcom*[26], *Chalice Tech*[27], *Digital/Compaq/HP*, *NEC*[28], *BMF*[29].
– *Speecys*[30]: Humanoider Roboter mit NetBSD & PowerPC.
– *Sonys* Network Stack der *Playstation Portable* und der Emotion-Engine–EEnet-Bibliothek basiert auf dem TCP/IP Stack von NetBSD[31].

Änderungen, Patches und Quellcode von externen Entwicklern werden über Mailinglisten, Problemberichte und persönliche Kontakte akzeptiert. Bei hinreichender Qualität des Quellcodes werden den Entwicklern Schreibrechte erteilt, wobei jedoch stets klar die Interessen des NetBSD-Projekts im Vordergrund stehen, auch bei NetBSD-Entwicklern, die von Firmen bezahlt werden.

4 NetBSD – Sicherheit frei Haus

Sicherheit sollte ein integraler Bestandteil moderner Betriebssysteme und als solcher kein Zusatzfeature, sondern eine Selbstverständlichkeit sein. Neben der Untersuchung, dem Dokumentieren und dem Verbessern von Code bezüglich aktueller Sicherheitslücken beschäftigen sich die NetBSD-Entwickler auch mit regelmäßigen Inspektionen des Codes, um potenzielle Sicherheitslücken zu finden und auszubessern. Fragen zur Sicherheit werden bei NetBSD vom *NetBSD Security Officer* und dem *NetBSD Security Alert Team* behandelt.

NetBSD hat *Kerberos 5 (Heimdal)*, *OpenSSL*, *OpenSSH* und *IPsec* für *IPv4* und *IPv6* in das System integriert. Zusätzlich sind alle Dienste prinzipiell nach der Installation abgeschaltet. Neben verschiedenen Programmen zur Sicherheit bietet NetBSD durch die saubere Implementierung kaum Angriffspunkte. Außerdem ist die Administration eines NetBSD-Systems aufgrund der strikten Trennung von Basissystem und Anwendungsprogrammen wesentlich einfacher und übersichtlicher als bei vielen anderen Betriebssystemen. All das zahlt sich aus: In den anerkannten Foren zum Thema „Sicherheit" überzeugt NetBSD konstant durch sehr wenige Sicherheitsprobleme.

23 http://www.sgi.co.jp/solutions/security/pdfs/viewranger_english.pdf
24 http://panasonic.co.jp/pcc/products/en/netwkcam/lineup/c10a.html
25 http://www.brains.co.jp, http://www.netbsd.org/Changes/1999.html#mmeye
26 http://www.broadcom.com/products/1250.html
27 http://www.simtec.co.uk/products/EB110ATX/intro.html
28 http://www.engadget.com/entry/1234000293053108/
29 http://www.bm-f.com/products/overview.html
30 http://www.speecys.com
31 http://www.feyrer.de/NetBSD/blog.html/nb_20060621_2109.html

Explizit sind in NetBSD verschiedene Sicherheitsmechanismen integriert die sicherstellen sollen, dass unerwünschte Änderungen und Eingriffe am System verhindert oder zumindest erkannt werden:

Security Advisories – Sicherheitshinweise: Wenn ernsthafte Sicherheitsprobleme in NetBSD gefunden und behoben werden, wird ein *Security Advisory* veröffentlicht, das das Problem beschreibt und einen Verweis auf die Lösung enthält. Diese Anweisungen werden neben der Veröffentlichung auch auf der Projektseite archiviert.

File flags und Kernel Security Level: *File flags* ermöglichen es den Benutzern oder dem Administrator, Dateien mit bestimmten *flags* zu markieren und so vor Manipulationen zu schützen. *Kernel Security Levels* schränken bestimmte Systemfunktionen ein und ermöglichen den Einsatz von *file flags*. Beispiel: Ist der *Security Level 2* aktiviert, so sind alle Datenträger nur lesbar verfügbar und können nicht mehr ein- oder ausgehängt werden. Der TCP/IP-Filter kann nicht mehr verändert werden, und die Systemzeit lässt sich nur noch vor-, aber nicht zurückstellen.

Dateimanipulationen erkennen: `mtree` ist ein Instrument, um den Ist-Stand eines Dateisystems mit einem Soll-Stand zu vergleichen. Eingesetzt wird es vor allem, um installierte Binärdateien gegen eine vorher spezifizierte Liste abzugleichen und um Manipulationen an Dateien aufzudecken. Dazu erstellt man einen Fingerabdruck eines Dateisystems, in dem Informationen zu den Dateien abgelegt werden. Dieser Fingerabdruck kann dann gegen das laufende System abgeglichen werden und hilft so zuverlässig beim Aufdecken von Veränderungen an Dateien (beispielsweise durch Würmer, *Rootkits* oder Ähnliches).

Trojaner aussperren mit veriexec: Der NetBSD-Kernel unterstützt mit `veriexec` ein System um, das Ausführen von manipulierten Binärdateien zu verhindern. Hierzu wird die Prüfsumme eines Programms in den Kernel geladen und beim Start des Programmes mit der aktuellen Prüfsumme verglichen. Wurde das Programm z. B. von einem Wurm, *Rootkit* oder Einbrecher verändert, so verweigert der Kernel die Ausführung.

Partitionen verschlüsseln mit cgd: Mit `cgd` kann man beliebige Partitionen (außer „/“ selbst) auf Blockebene verschlüsseln. Ohne Angabe des Passworts hat niemand, auch nicht der Inhaber eines Administratorpasswortes oder jemand mit physikalischem Zugriff auf die Festplatte, eine Chance, an die Daten heranzukommen. Mit `cgd` kann man auch die Swap- und Temp-Partitionen verschlüsseln, um zu verhindern, dass dort gespeicherte geheime Daten öffentlich werden. Nach „oben“ verhält sich `cgd` wie eine Festplatte und kann mit jedem beliebigen Dateisystem versehen werden.

System calls kontrollieren mit systrace: Niels Provos' *systrace* erlaubt es, einzelne Systemaufrufe eines Programms zu kontrollieren. So ist es beispielsweise möglich, den *Apache* von einem normalen Benutzerkonto aus zu starten, weil dieser Benutzer via *systrace Apache* an den Port 80 binden darf – so dass *Apache* selbst nicht mehr mit Root-Rechten läuft.

Tägliche Sicherheitsüberprüfung: Die beiden Shellskripte „/etc/daily" und „/etc/security" erlauben es, das gesamte System auf Sicherheitslücken hin zu untersuchen. Sie können zu bestimmten Zeiten von *cron* gestartet werden und generieren einen umfassenden Bericht zu Sicherheitsproblemen.

Software auf Sicherheitslücken prüfen: Durch das Paket *audit-packages* können die installierten Pakete mit einer vom NetBSD-Projekt gepflegten Liste von bekannten Sicherheitslücken abgeglichen werden. So werden alle Pakete mit Sicherheitsproblemen aufgelistet und können dementsprechend aktualisiert werden.

Paketfilter und Firewalling: Mit *IPFilter* und *PF* unterstützt NetBSD im Basissystem zwei ausgereifte Paketfilter, die jedes NetBSD-System zum Betrieb einer ausgereiften und stabilen Firewall befähigen.

Umfangreiche Sicherheitspakete: Mit *pkgsrc* lassen sich viele weitere wichtige Sicherheitspakete installieren. Dazu zählen beispielsweise *snort*, *AIDE*, *Tripwire*, *CFS*, *chkrootkit*, *Nessus*, *Amap*, *GnuPG* und *honeyd*.

5 Aktuelle Applikationen mit pkgsrc

5.1 Herausforderungen bei der Softwareinstallation

Die Installation von Open-Source-Software unter *UNIX* ist generell mit einigen Problemen behaftet: Zum einen gibt es in der Regel eine Fülle von unterschiedlichen Softwarelösungen, die darüber hinaus häufigen Versionswechseln unterworfen sind. Zum anderen ist das Kompilieren von Quellcode teilweise zeitaufwändig und fehlerbehaftet. Zusätzlich kann auch die mangelnde Portabilität einer Software zu Problemen führen.

Hinzu kommt, dass Software-Installationen aus dem Quellcode heraus selbst für Experten oft eine Herausforderung sind: Der Anwender muss über ein solides Grundwissen über die benötigten Werkzeuge verfügen, außerdem existieren meist verschiedene Arten von Konfigurationen, die alle ihre jeweiligen Eigenheiten haben. Hinzu kommen die Abhängigkeiten von anderen Programmpaketen. Sollte dann einmal ein Fehler bei der Installation auftreten, wird zu dessen Behebung oft nicht vorhandenes Expertenwissen benötigt.

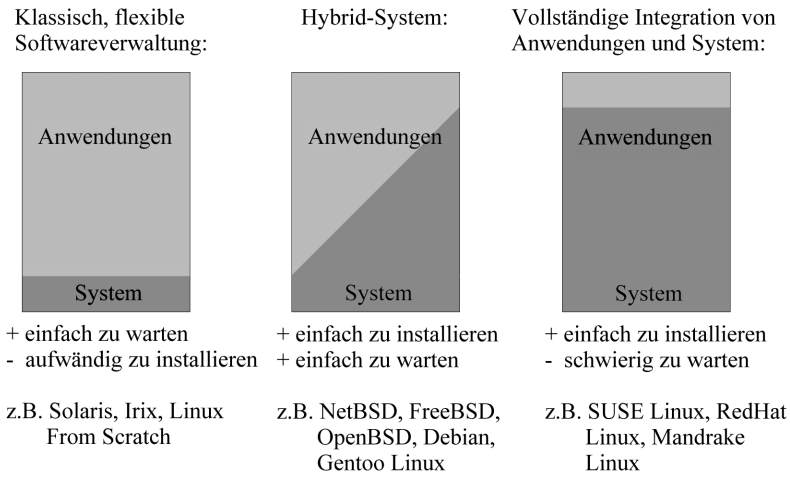


Abbildung 2: Aufwand verschiedener Konzepte für die Installation und Wartung von Betriebssystemen

5.2 Lösungen: Je nach Umgebung!

Die Lösung der im vorigen Abschnitt geschilderten Probleme kann je nach Anforderungen und Umgebung anders aussehen: Falls sich Software wenig ändert (z. B. bei vorgefertigten Binärdistributionen), empfehlen sich automatisierte Installationsroutinen. Dieser Ansatz wird vor allem bei Desktop-Systemen wie *Windows* oder *SUSE Linux* verwendet, wobei einfachere Installation mit komplexerer Wartbarkeit erkauft wird. Steht der Wunsch nach einfacher Wartung im Vordergrund (z. B. bei Webservern mit *Solaris*, *Apache* und *PHP*) empfiehlt es sich, ein stabiles Grund–Betriebssystem und wichtige Pakete selbst zu kompilieren. Das Ergebnis ist ein leicht wartbares System, welches jedoch mehr Arbeit beim Einrichten erfordert. Eine ideale Lösung aus beiden Welten stellen Systeme dar, die sowohl eine einfache Installation als auch flexible Wartbarkeit bieten. Ein Beispiel für eine derartige Lösung ist das von NetBSD verwendete *pkgsrc*[32].

Mit Hilfe des Paketsystems *pkgsrc* können mittlerweile über 6 000 frei erhältliche Software-Pakete (z. B. *KDE*, *GNOME*, *XFCE*, *OpenOffice.org*, *Apache*, *PostgreSQL*, *Firefox*, *Samba*, LaTeX etc.) ohne Aufwand installiert werden. Fertig kompilierte Pakete können von CD, DVD oder FTP-Servern installiert werden, alternativ können Pakete auch mit Hilfe von *pkgsrc* mit einem einfachen „make install“ selbst kompiliert werden.

32 Siehe auch http://www.pkgsrc.org.

5.3 pkgsrc als portable Lösung

Das Paketsystem von NetBSD – *pkgsrc* – stellt für die genannten und eine Reihe weiterer Probleme Lösungsmöglichkeiten bereit:

- *pkgsrc* ist ein System zum einfachen Installieren und Updaten von Paketen.
- *pkgsrc* ist ein Source-basiertes Paketverwaltungssystem.
- *pkgsrc* verwendet Original-Sourcecode zum Kompilieren.
- *pkgsrc* bietet die Möglichkeit, Binärpakete zu erzeugen und zu installieren.
- *pkgsrc* besteht aus zwei Komponenten, den Verwaltungstools und der eigentlichen Paketsammlung.
- *pkgsrc* behandelt Abhängigkeiten automatisch.

Ursprünglich von *FreeBSD* auf NetBSD portiert, hat *pkgsrc* heute als primäre Entwicklungsplattform NetBSD, wobei das System jedoch für viele weitere Systeme portiert wurde und auf diesen läuft: Unter anderem wurde *pkgsrc* für *AIX*, *BSD/OS*, *Darwin*, *FreeBSD*, *Irix*, *NetBSD*, *OpenBSD*, *Solaris* und *Dragonfly BSD* sowie auf die Linux Distributionen *SUSE*, *Debian*, *ROOT Linux*, *Slackware*, *RedHat*, *Fedora*, und *Mandrake* portiert.

5.4 pkgsrc in der Praxis

Die Installation von Paketen mittels *pkgsrc* ist denkbar einfach, um z. B. die grafische Benutzeroberfläche *KDE* mit all ihren Abhängigkeiten zu installieren, reichen folgende Befehle:

- cd pkgsrc/meta-pkgs/kde3
- make install

pkgsrc kann größtenteils ohne Root-Rechte betrieben werden, selbst umfangreiche Pakete wie *Firefox* oder *KDE* funktionieren ohne diese. Weiterhin kann das Verzeichnis, in das alle Pakete installiert werden sollen, frei gewählt werden. Die Standardeinstellung von „/usr/pkg“ ist für eine systemweite Installation sinnvoll, bei einer privaten Installation von Paketen ohne Root-Rechte ist oft das Heimatverzeichnis des Anwenders die bessere Wahl.

Neben dem Kompilieren von Quellcode – was sehr lange dauern kann – unterstützt *pkgsrc* auch das Erstellen von vorkompilierten Binärpaketen, die dann einfach mit *pkg_add(8)* installiert werden können.

Da Sicherheit auch bei *pkgsrc* groß geschrieben wird, ist mit dem Paket *audit-packages* die Möglichkeit vorhanden, über Sicherheitslücken und damit verbundene Updates informiert zu werden, um dann die entsprechenden Paketinstallationen zu aktualisieren.

Viermal jährlich wird ein Release von *pkgsrc*[33] erstellt, bei dem besonders darauf geachtet wird, dass alle Pakete optimal aufeinander abgestimmt sind; sie werden dann auch für die diversen Plattformen als fertige Binärpakete bereitgestellt.

6 Bleiben Sie am Ball!

pkgsrc wächst und gedeiht als Nebenprojekt von NetBSD seit 1997, NetBSD selbst seit März 1993 – länger als jede andere Alternative im Open-Source-Bereich.

Dank der Arbeit der Entwickler und dem Engagement vieler Anwender ist NetBSD heute stärker denn je, und die Entwicklungen gehen weiter: Die Unterstützung für Virtualisierungstechnologien wie *Xen* wird ständig verbessert, an verschiedenen neuen Dateisystemen wird intensiv gearbeitet, die NetBSD-Sicherheitsarchitektur wird weiter ausgebaut, und System- und Sicherheits-Updates werden zukünftig einfacher werden.

Das NetBSD-Projekt hat weiterhin eine interessante und starke Zukunft vor sich. Anwender haben die Gewissheit, dass ihr Betriebssystem von erfahrenen Programmierern weiterentwickelt wird und dass Innovationen beständig in das System einfließen. Weil das NetBSD-Projekt offen für Ansätze und Lösungen aus anderen Open-Source–Projekten ist, behalten die Anwender den Anschluss an die besten Entwicklungen der freien Software.

Hilfe bei Problemen ist schnell und unbürokratisch über Mailinglisten, IRC-Channel und über die Fehlerdatenbank möglich. Für formellere Unterstützung stehen auch die auf der NetBSD-Webseite aufgelisteten professionellen Berater zur Verfügung. Auch wenn keine Servicetelefonnummer existiert, werden Fragen vom NetBSD-Team selbstverständlich beantwortet – ohne dass man in einer Warteschleife hängt.

33 Zu finden unter http://ftp.netbsd.org/pub/pkgsrc/.