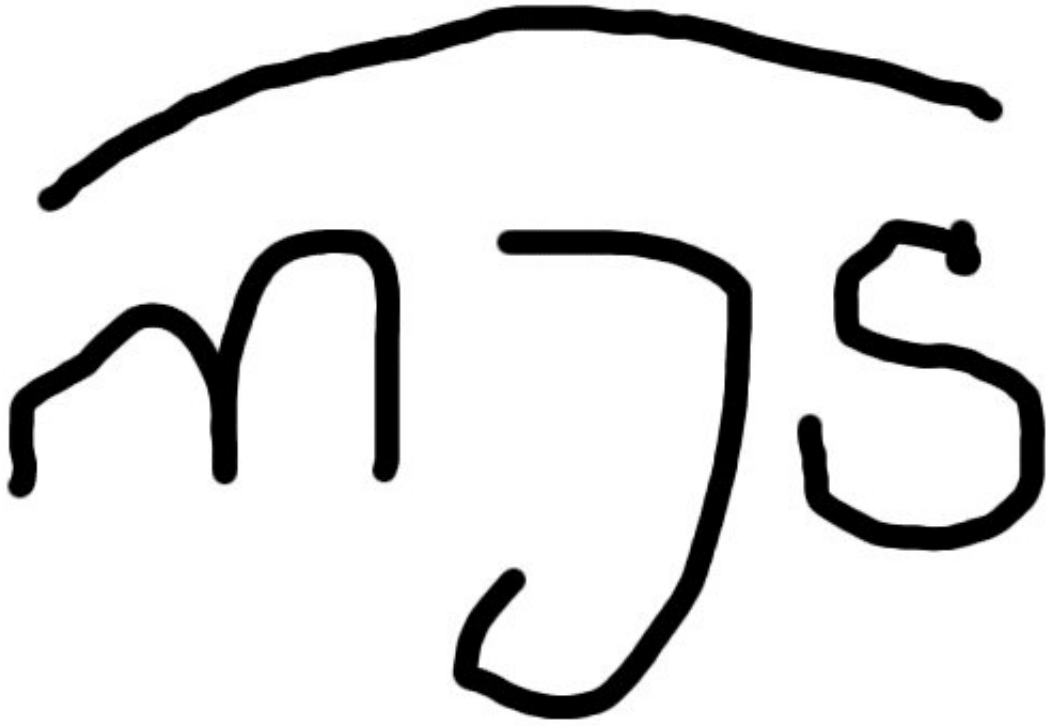

Magdeburger Journal zur Sicherheitsforschung

Gegründet 2011 | ISSN: 2192-4260
Herausgegeben von Stefan Schumacher und Jan W. Meine
Meine Verlag – Wissenschafts-, Sach- und Fachbuchverlag, Magdeburg

Sicherheit durch Freiheit und Offenheit? Ein Fallbeispiel.

Dr. Hubert Feyrer

Nach der Definition zentraler Begriffe wird das NetBSD Projekt mit seiner Governance sowie den Prozessen für Entwicklung und Sicherheit vorgestellt. Die Prozesse und ihre Ergebnisse werden vermessen, und die Fragestellung, ob Sicherheit durch Freiheit und Offenheit möglich ist, wird basierend auf dem vorliegenden Projekt beantwortet.

Zitationsvorschlag: Feyrer, H. (2012). Sicherheit durch Freiheit und Offenheit?: Ein Fallbeispiel. *Magdeburger Journal zur Sicherheitsforschung*, *2*, 202 – 216. Zugriff am 16. Juli 2012, unter `http://www.wissens-werk.de/index.php/mjs`

Intro

In der öffentlichen Diskussion gewinnen aktuell »offene« Projekte an breitem Zuspruch - dies gilt insbesondere für Open Source, Open Access und Offene Standards[1]. Gleichzeitig ist oft der Ruf nach dem Erhalt von Freiheit zu hören[2]. Der vorliegende Artikel befasst sich damit, ob auch Sicherheit durch einen »offenen« Ansatz umsetzbar ist, der gleichzeitig Freiheit gewährleistet. Aufgezeigt wird dies an Fallbeispiel des Open Source Betriebssystems NetBSD.

Definitionen

Der Ursprung des Wortes »Sicherheit« liegt im Lateinischen, und kann von dort als »sorglos« bzw. »ohne Sorge« abgeleitet werden[3]. Der Zustand von Sicherheit ist dabei nicht absolut, sondern relativ zum für tragbar empfundenen Risikoniveau – die ideale Ausprägung liegt bei 100%, dies ist jedoch auch graduell zu erreichen. Das Thema hat viele Dimensionen[4], wobei hier die technisch-kulturelle relevant ist. Der Technik-Bezug liegt einerseits bei einem Software-Projekt und andererseits dem dabei entwickelten Betriebssystem, kulturell relevant ist hier der zum Produkt führende Entwicklungs-Prozesses sowie die unterstützenden Sicherheitsprozesse die von der Entwicklergemeinde umgesetzt werden.

Bei Sicherheit im Umfeld von Computern und Informationstechnologie wird schnell der Begriff »IT-Sicherheit« bemüht. Breiter und faktisch korrekter ist hier »Informationssicherheit« gemeint, da die »Sorgen« im Umfeld von Computern und Informationen oft nicht (nur) durch Informations-Technik adressiert werden, sondern oft durch entsprechende organisatorische Vorgaben und deren Befolgung[5].

Der Begriff der »Freiheit« bzw. das Attribut, »frei« zu sein wird heute oft doppeldeutig benutzt. Freiheit wird oft als nicht-vereinbar bzw. Gegenteil von »Sicherheit« angesehen, da mit »Sicherheit« einhergehende Regelungen als Einschränkung der Freiheit verstanden werden[6]. Zudem kommt, dass mit »frei« oft auch die kostenlose, uneingeschränkte Verfügbarkeit, etwa durch Urheberrechtliche Auflagen wie Lizenzforderungen, verbunden sind[7].

Wenn hier von »Offenheit« gesprochen wird, so ist dies im Sinne von Offenlegung, Transparenz und Zugänglichmachung gemeint – Open Access und Open Source sind entsprechende Beispiele. Genau wie beim Begriff der »Freiheit« wird hier oft auch ein Zugang ohne nennenswerte urheberrechtliche Einschränkungen oder Auflagen verstanden, nicht jedoch dass per se von einer Kostenfreiheit bzw. kostenlosem Erhalt ausgegangen werden kann[8,9].

Oft werden »Freiheit« und »Offenheit« im Umfeld quelloffener Software zusammen genannt, um zu unterstreichen dass einerseits keine Kosten bei der Beschaffung involviert sind, und auch dass das

1 Mundhenke (2007) S. 69ff
2 Mill (2011) S. 78ff
3 Wikipedia (2012)
4 Christopher Daase (2010)
5 Hans Halstrick and Karsten M. Decker (2009)
6 Hans Peter Bull (2011) S. 8f
7 Urs Fässler (2011)
8 Magnus Bergquist and Jan Ljungberg and Bertil Rolandsson (2011)
9 Michael Schwarz and Yuri Takhteyev (2010)

Produkt ohne inakzeptablen Einschränkungen benutzt, verändert und weitergegeben werden darf.

Im Folgenden wird die Wechselwirkung von Freiheit und Offenheit auf die Sicherheit des Open Source Betriebssystems NetBSD aufgezeigt.

Was ist NetBSD?

NetBSD ist ein quelloffenes Betriebssystem. Als 1993 an der Universität von Berkeley in Kalifornien das Budget für deren Betriebssystem-Projekt »BSD-Unix« gestrichen wurden, hat man überlegt, wie man das Projekt weiterführen könnte. Aufgrund des langsam an weltweiter Bedeutung für das Zielpublikum gewinnenden Internets entschied man sich, dies als neues Projekt »NetBSD« zu tun. Der Fokus war, ein offenes, frei verfügbares Betriebssystem zur Verfügung zu stellen, das für Privatanwender, Forschungseinrichtungen und Firmen kostenfrei zur Verfügung steht, und das technisch auf einer möglichst breite Palette an Rechner-Systemen läuft.

»Quelloffen« – im Englischen »Open Source« – heisst an dieser Stelle, dass nicht nur die fertig installierbare, maschinenlesbare Binärversion zur Verfügung stellt. Stattdessen ist auch der menschenles- und -änderbare Quellcode des Systems verfügbar. Interessierte können so Einblicke in die internen Abläufe erhalten und verstehen, aber auch Änderungen machen und darauf basierende Änderungen selbst in Maschinencode überführen und laufen lassen – ohne dass das NetBSD Projekt involviert ist. Auf diese Art ist es möglich, eigene Entwicklungen zu machen und weiterzuverteilen, aber auch Fehler zu beheben ohne auf einen Hersteller angewiesen zu sein[1].

Viele technische Entwicklungen wurden im Bereich der Betriebssysteme im Verlauf der letzten Jahrzehnte gemacht, und entsprechend wurde auch NetBSD von einer weltweit verteilten Entwicklergemeinde konstant weiterentwickelt und zur Verfügung gestellt. Die vielen Anwendungsmöglichkeiten und Attribute hier aufzuzählen würde den Rahmen sprengen, daher sei hier auf die Webseite des Projektes verwiesen[2] .

In den Zielen des NetBSD Projektes sind Freiheit und Offenheit genannt. Freiheit und Offenheit sind hier in Sinne eines uneingeschränkt zugänglichen Systems zu verstehen. Das Betriebssystem ist sowohl in ausführbarer Form als auch im Quellcode uneingeschränkt verfügbar, Änderungen können gemacht werden und die Lizenz erlaubt es Firmen auch, geistiges Eigentum das durch Weiterentwicklungen geschaffen wurde zu schützen.

Sicherheit ist eines der explizit erklärten Ziele des NetBSD Projektes, was auch eine Grundvoraussetzung jedes zeitgenössischen Betriebssystems ist, so auch von NetBSD[1].

Die folgenden Abschnitte zeigen, wie Sicherheit im Rahmen des NetBSD Projektes durch die Entwickler umgesetzt wird. Hier werden vordergründig die für die Produktentwicklung relevanten technischen und organisatorischen Aspekte der Informationssicherheit dargestellt. Im Anschluss wird das erreichte Ergebnis einerseits gegen den den internationalen Standard zum Management von Informationssicherheit verglichen, und

1 Aho, Lam, Sethi und Ullman (2006)

2 The NetBSD Foundation (2012m)

1 The NetBSD Foundation (2012m)

dem gegenüber die Anzahl der Sicherheitsprobleme im Endprodukt gestellt.

Governance

Wie eingangs beschrieben ist das NetBSD Projekt überall dort zuhause, wo es Internet gibt, und entsprechend ist die Anwender- und Entwicklergemeinde des Systems weltweit verstreut. Als zentrale Instanz, die sich um die rechtlichen Geschäfte kümmert existiert eine in den Vereinigten Staaten von Amerika eingetragene Stiftung, The NetBSD Foundation (TNF). Diese nimmt Spendengelder entgegen, bezahlt Rechnungen für Server-Hardware, Internet-Hosting und Auftrags-Entwicklungen. Desweiteren ist die NetBSD Stiftung auch rechtlicher Eigentümer des NetBSD Betriebssystems. Sie sorgt dafür, dass möglichst viel des NetBSD Codes einheitlich unter der NetBSD Lizenz steht, und dass entsprechend auch nur ein einziger Ansprechpartner existiert. Dies vereinfacht die Kommunikation mit möglichen Lizenznehmern bei Rückfragen enorm.

Die NetBSD Stiftung ist in mehrere »Komitees« unterteilt, die sich um die technische Weiterentwicklung des Betriebssystems (core), Integration von Open Source Anwendungen (pkgsrc-pmc), Mitgliederverwaltung (membership-exec) und Finanzen (finance-exec) kümmern. All dies unter der Aufsicht des »Board of Directors« (board). Abbildung 1 zeigt das zugehörige Organigramm[2].

Die Organisation des Boards und der einzelnen Komitees wird am Ende von einzelnen Mitgliedern der Stiftung gestellt. Diese unterschreiben mit ihrem Namen auf dem Mitgliedsantrag dafür, dass sie die Ziele der Stiftung vertreten[3]. Zusätzlich zum »Paper Trail« wird von jedem Anwärter auf die Mitgliedschaft in der NetBSD Stiftung auch erwartet, dass er einen PGP-Schlüssel besitzt, der es ermöglicht verschlüsselte und digital unterschriebene E-Mail zu verschicken. Dieser Schlüssel muss von mindestens einem bestehenden Mitglied der NetBSD Stiftung digital unterschrieben sein[1]. Dies setzt ein Treffen außerhalb des Internets voraus, und stellt sicher dass die Identität des Anwärters mittels Ausweis geprüft wird. Die PGP-Schlüssel bilden ein Netzwerk, das sogenannte NetBSD »Web of Trust«. Mithilfe des Netzwerks können zum einen fälschungssichere Mitteilungen an Administratoren gesandt werden, um Account-Details wie Passwörter zu setzen. Darüber hinaus kann der frisch signierte Schlüssel dann verwendet werden, um wiederum Schlüssel neuer Mitglieder zu signieren, und so die Entwicklergemeinde zu vergrößern. Nicht zuletzt kann PGP eingesetzt werden, um verschlüsselte Nachrichten auszutauschen, was zur Wahrung von Vertraulichkeit und Integrität notwendig ist, etwa bei der Durchführung von Wahlen des Boards[2].

Mit der Aufnahme in die NetBSD Stiftung erhalten Mitglieder auch Schreibzugriff auf den NetBSD Quellcode und werden damit zu offiziellen Entwicklern, die Änderungen direkt einstellen können. Ein oder mehrere Mentoren – genannt »Sponsor« – stehen hier unterstützend sowohl bei der Antragsprozedur als auch bei Fragen zum Entwicklungsvor-

2 The NetBSD Foundation (2012l)

3 The NetBSD Foundation (2012i)

1 The NetBSD Foundation (2012j)

2 The NetBSD Foundation (2012j)

gehen mit Rat und Tat zur Seite. Nicht zuletzt obliegt es auch den Mentoren, die Qualitätskontrolle der Entwicklungsarbeiten auszuführen.

Entwicklungsprozess

Woher kommen neuer Code bzw. Entwicklungen im NetBSD Betriebssystem? Hier gibt es mehrere mögliche Quellen[3]. Die bestehenden Entwickler, die aufgrund der Mitgliedschaft bei der NetBSD Stiftung gleichzeitig Schreibzugriff auf das CVS-Repository haben, wurden bereits genannt.

Als langjähriges Open Source Projekt besitzt NetBSD auch eine rege Community von Anwendern und Interessenten[1]. Sie selbst besitzen zwar keinen direkten Schreibzugang zum Quellcode, dieser ist jedoch – ganz im Sinne des ursprünglichen BSD Unix – frei zum Download erhältlich. Interessierte können den Quellcode per FTP und anonymem CVS komplett erhalten und so als Basis für eigene Entwicklungsarbeiten nutzen. Für Studienzwecke ist der Quellcode außerdem via Webbrowser lesbar[2].

Verbesserungen und neue Entwicklungsarbeiten können ohne Schreibzugriff auf den NetBSD Quellcode gemacht werden. Erst für die eigentliche Integration in den offiziellen Quellcode ist dieser nötig, und oft wird hier zwischen Community und Entwicklern zusammengearbeitet, um durch Peer-Review die Qualität und Sicherheit der Entwicklungsarbeiten zu garantieren. Erst wenn alle gängigen Standards erfüllt sind kann der Code in das NetBSD CVS-Repository übernommen werden.

Als letzter Weg, NetBSD weiterzuentwickeln bestehen gezielte Projekte, die einzelne Teile verbessern oder neue Subsysteme erstellen. Dies geschieht zum Beispiel durch den jährlich stattfindenden Google »Summer of Code«, oder auch durch Arbeiten die von der NetBSD Stiftung direkt aus den Spenden bezahlt werden.

Sicherheitsprozess

Bei einem komplexen Softwareprodukt wie einem Betriebssystem, das aus zahlreichen einzelnen Komponenten besteht, bleibt es nicht aus, dass Sicherheitsprobleme zutage treten. Das NetBSD Projekt hat eine Struktur etabliert, um dem entgegenzuwirken.

Eine Voraussetzung um Sicherheitslücken zu schließen ist, dass diese bekannt werden. Ein Team von »Security Officers« ist hier über Mailing Listen und private Kontakte mit Sicherheitsforschern, Herstellern und Computer Emergency Response Teams (CERTs) in Kontakt, um sofort über neu entdeckte Probleme unterrichtet zu sein. Es wird auch eine eigene Kontakt-Adresse bereitgehalten, mittels derer das Sicherheitsteam angeschrieben werden kann. Da hier oft vertrauliche Informationen ausgetauscht werden macht sich die bestehende PGP-Infrastruktur bezahlt.

Ist ein Sicherheitsproblem bekannt, wird es mit Hilfe der zuständigen und weiterer fachkundigen Entwicklern analysiert, und eine optimale Lösung gefunden. Bevor Änderungen am Quellcode gemacht werden wird sichergestellt, dass keine sonstigen Funktionalitäten negativ beeinflusst werden. Dies kann oft einige

3 The NetBSD Foundation (2012a)

1 The NetBSD Foundation (2012c)

2 The NetBSD Foundation (2012b)

Zeit in Anspruch nehmen. Wenn das gefundene Problem Angriffe über das Netz erlaubt und Datenspionage oder Denial-of-Service-Angriffe möglich sind, dann wird oft in einem schnellen Zwischenschritt eine erste Lösung angebracht, die dann weiter ausgearbeitet wird. Dies ist vor allem für die verschiedenen Server relevant, die die Infrastruktur des NetBSD Projektes darstellen, und die natürlich alle selbst unter NetBSD laufen - CVS-Repository, Webserver, E-Mail-Server und viele weitere[3].

Im Verlauf der Analyse und beim Abstellen der Verletzbarkeit des Quellcodes wird darauf geachtet, dass noch keine Änderungen in CVS eingestellt werden, da dies möglichen Angreifern Hinweise auf die Angriffspunkte geben kann. Denn nur weil der NetBSD-Quellcode die Sicherheitslücke nicht mehr hat existieren nach wie vor laufende Systeme, die erst noch aktualisiert werden müssen!

Um diesen wichtigen Schritt einzuleiten wird ein »Security Advisory« veröffentlicht, das den Sachverhalt darstellt und Schritte zur Abhilfe beschreibt. Diese Ankündigung wird dann auf eigenen Security-Mailing Listen sowie auf dem NetBSD Webserver veröffentlicht. Zeitgleich wird dann auch der Quellcode für den Patch im CVS-Repository abgelegt – ein Schritt der auch mit einer E-Mail auf der `source-changes` Mailing-Liste verfolgt werden kann[4].

Das NetBSD Projekt stellt Patches im Quellcode für die aktuelle Entwicklungsversion (»NetBSD-current«) bereit, außerdem werden Sicherheits-Patches auch für den aktuellen und den vorangegangenen Betriebssystem-Release an einem eigenen Zweig (»Branch«) im Quellcode gepflegt. Administratoren können so die Sicherheit ihrer Systeme in einem ansonsten stabilen Betrieb sicherstellen[5].

Die Dokumentation dieses Prozesses ist auf den Webseiten des NetBSD Projektes zu finden[6,7,8], dort ist auch ein Archiv aller bisher veröffentlichten Security Advisories zu finden[1]. Als Diskussionsforum sei weiterhin die tech-security Mailing Liste mit ihrem Archiv genannt[2].

Abgrenzung: Was NetBSD nicht hat bzw. nicht offenlegt

Sicherheit ist immer als Prozess zu sehen. Der vorliegende Ablauf für die Behandlung von Sicherheitslücken im Rahmen der Betriebssystem-Entwicklung von NetBSD zeigt dies. Dem gegenüber sollen hier zwei Themenbereiche abgegrenzt werden, die oft in anderen Betriebssystem-Projekten gefunden werden, bei NetBSD jedoch aktuell nicht anzutreffen sind.

Rein technisch ist hier das Thema »Code-Signing« zu nennen. Primär zur Verteilung von vorcompiliertem Maschinencode interessant wird damit sichergestellt, dass nur offiziell freigegebene Patches verteilt und installiert werden. Durch die Signatur des Codes mit einer digitalen Unterschrift wird versucht, Manipulationen durch unberechtigte Dritte auszuschließen.

NetBSD verteilt aktuell keine Patches

3 The NetBSD Foundation (2012h)

4 The NetBSD Foundation (2012h)

5 The NetBSD Foundation (2012g)

6 The NetBSD Foundation (2012h)

7 The NetBSD Foundation (2012k)

8 The NetBSD Foundation (2012g)

1 The NetBSD Foundation (2012f)

2 The NetBSD Foundation (2012e)

im Binärformat, so dass dieses Thema hier nicht anzuwenden ist. Dies ist zum einen durch die große Anzahl verschiedener Hardware-Plattformen (und damit bereitzustellender Patches) begründet. Zum Anderen bedarf dies neuer Infrastruktur, die ihrerseits neue Angriffsvektoren eröffnet, wie Angriffe des Flame-Wurms zeigen, der sich als Windows-Update-Server ausgibt[1]. NetBSD setzt hierzu bei der Verteilung der offiziellen Release-Binaries auf digitale Prüfsummen, Patches werden mittels per SSH abgesichertem CVS verteilt - ein Weg der bis Dato als sicher gilt.

Neben technischen Massnahmen, um die Sicherheit und Auslieferung des NetBSD Betriebssystems zu gewährleisten, bieten sich eine Reihe organisatorischer Massnahmen an. Standards wie BSI Grundschutz oder ISO 27001 können hier als Richtschnur dienen. Diese bieten eine Reihe von »Best Practices«. Intern sind diese Vorgehen bei Entwicklern und Administratoren des NetBSD Projektes zumindest in Teilen sehr wohl vorhanden, aber aus Sicherheitsgründen sind diese Angaben nicht offengelegt. Da sie nicht durchgängig umgesetzt sind kann hier auch keine Zertifizierung nachgewiesen werden. Aber auch ohne diese ergibt sich die Verpflichtung zum Schutz des Betriebssystems, letztendlich über die NetBSD Stiftung als Rechte-Halter. Und entsprechend ist auch der höchste Schutzbedarf klar festgelegt: Das CVS-Repository mit dem NetBSD-Quellcode.

Bestandsaufnahme

Mit den bisherigen Ausführungen wurde die Governance des NetBSD Projektes aufgezeigt, mit besonderem Fokus auf den Entwicklungsprozess und die Behandlung des Themas Sicherheit hinsichtlich des Endproduktes. An dieser Stelle soll nun die Frage erörtert werden, ob das Ergebnis dem Anspruch eines modernen, sicheren Betriebssystems gerecht wird.

Um dies zu beurteilen werden zwei Metriken herangezogen: Zum einen ein Vergleich des Entwicklungsprozesses gegen den entsprechenden Abschnitt des internationalen Standards für das Management von Informationssicherheit, der ISO 27001 (»Information Security Management System«, ISMS)[2], zum anderen eine Darstellung der gefunden Sicherheitslücken in den vergangenen Jahren im Vergleich zu ähnlichen Produkten und Projekten.

Abgleich: NetBSD und die ISO 27001

Der Standard für Management-Systeme zur Informationssicherheit ISO 27001 beschreibt zum einen ein Management-System, zum anderen fordert er im normativen Anhang die Umsetzung von über Hundert verschiedenen Massnahmen zur Steigerung der Sicherheit. Im Anhang A.12 ist Sicherheit für »Beschaffung, Entwicklung und Wartung von Informationssystemen« abgedeckt, und gegen diese soll nun das bei NetBSD eingesetzte Verfahren verglichen werden.

Die NetBSD Homepage beschreibt NetBSD als »a free, fast, secure, and

1 heiseSecurity (2012)

2 ISO/IEC (2005)

highly portable Unix-like Open Source operating system.« Damit ist der Anspruch an Sicherheit klar und explizit dargelegt, die nachgelagerten Prozesse setzen dies wie aufgezeigt um[3]. Die korrekte Verarbeitung einzelner Komponenten wird durch Regressionstests sichergestellt[4]. Als in den USA beheimatetes Projekt hat NetBSD seit jeher auf die Exportbestimmungen der USA zu achten, dies ist auch nach wie vor so[5,6].

Um die Sicherheit von Systemdateien zu garantieren werden die Server des NetBSD-Projektes adäquat technisch abgesichert, es greift ein striktes Berechtigungskonzept und Sicherheitspatches werden wie oben beschrieben zeitnah und mit Bedacht eingespielt. Der Schreibzugriff auf den Quellcode ist ebenfalls streng reglementiert und nur für registrierte Entwickler freigeschaltet[7].

Die Sicherheit bei Entwicklungs- und Unterstützungsprozessen ist ebenfalls gegeben. Änderungen am Quellcode werden per CVS nachverfolgt und auf der source-changes Mailing Liste protokolliert. Entwicklungstätigkeiten werden im Entwurf diskutiert und größere Änderungen sind vom Core Team als technische Leitung freizugeben. Extern ausgelagerte Entwicklungsarbeiten werden per Mailing Liste oder in Form eines »Problem Report« als Ticket eingereicht und auch nur nach intensivem Review integriert[8].

3 ISO/IEC (2005) A.12.1

4 ISO/IEC (2005) A.12.2

5 The NetBSD Foundation (2012d)

6 ISO/IEC (2005) A.12.3

7 ISO/IEC (2005) A.12.4

8 ISO/IEC (2005) A.12.5

Last but not Least ist über den oben beschriebenen Sicherheitsprozess sichergestellt, dass Informationen über Schwachstellen frühzeitig bekannt und entsprechend behandelt werden[9].

Zusammenfassend kann gesagt werden, dass die Vorgaben der ISO 27001 für die Entwicklung und Wartung der Informationssicherheit wie hier beschrieben erfüllt werden. Damit sind gute Voraussetzungen für die Entwicklung eines Betriebssystem gelegt, das zeitgemäßen Anforderungen bezüglich Sicherheit entspricht.

Ein vollständiger Abgleich aller Normziele der ISO 27001 würde den Rahmen dieser Abhandlung sprengen, und auch mangels öffentlich verfügbarer Informationen nicht möglich sein.

Abgleich: Schwachstellen von NetBSD und andere Betriebssysteme

Nach dem theoretischen Vergleich der erfüllten Sicherheitsanforderungen der ISO 27001 soll nun das Endergebnis selbst betrachtet werden: Wieviele Sicherheitslücken wurden bei NetBSD in den letzten Jahren bekannt? Als Vergleichsbasis wird die Anzahl der Common Vulnerabilities and Exposures (CVE) Einträge in der National Vulnerability Database des amerikanischen National Institute of Standards and Technology (NIST) benutzt[10], es erfolgt eine Darstellung von gefundenen Schwachstellen seit 1998.

Die Liste der betrachteten Betriebssysteme ist relativ lange, was mit der breiten Anzahl an Anwendungsmöglichkeiten von NetBSD und damit den entspre-

9 ISO/IEC (2005) A.12.6

10 National Institute of Standards and Technology (2012)

chenden Mitbewerbern auf dem jeweiligen Markt begründet ist. Zum einen sind dies historisch die weiteren BSD-Projekte FreeBSD und OpenBSD, Linux als weiteres »großes« Open Source Betriebssystem, und Microsoft Windows und Apple Mac OS X als weitverbreitete Systeme für den allgemeinen Einsatz vom Desktop bis zum Server. Als Pendant im Server-Einsatz werden die klassischen Unix-basierten Systeme Oracle Solaris, IBM AIX und HP's HP-UX betrachtet. Da NetBSD in diversen mobilen Systemen zum Einsatz kommt wurden Apple iOS, Google Android, RIM Blackberry und Nokia Symbian hinzugenommen. Last but not least ist eine der besonderen Stärken von NetBSD der Einsatz im Embedded Umfeld, weshalb hier speziell QNX, VxWorks und Siemens Simatic – bekannt durch die Steuerung der Uranzentrifugen im Iran – gelistet.

Die gesammelten Daten sind in den Tabellen 1, 2 und 3 gelistet, Tabelle 4 zeigt dieselben Daten, allerdings für jedes Betriebssystem über alle Jahre kumuliert.

Abbildung 2 zeigt den zeitlichen Verlauf der gefundenen Sicherheitslücken für die einzelnen Betriebssysteme im Vergleich. Abbildung 3 enthält dieselbe Darstellung, allerdings wurde hier die »Spitzenreiter« mit den meisten Schwachstellen – Linux, Mac OS X und Solaris – ausgeblendet, um ein klareres Ergebnis des Feldes zu erhalten. Abbildung 4 verfeinert die Darstellung, indem nur die vier Systeme gezeigt werden, die über den gesamten Betrachtungszeitraum die niedrigsten und damit besten Werten nachweisen.

Hier nun die Beobachtungen, die aus den gelisteten Daten und ihrer grafischen Darstellung gemacht werden können. Als erstes zeigt der Vergleich aller Systeme in Abbildung 2 klar die Spitzenreiter Linux, Windows und Mac OS X. Dies ist naheliegend, da diese als Desktop-Systeme eine breite Funktionalität anbieten, die auch entsprechend angreifbar ist.

Trends in Abbildung 3 zeigen den Anstieg der Marktverbreitung von Android, Apples iPhone und Mac OS X. Gleichzeitig ist ein Abstieg der in den BSD-Projekten gefundenen Sicherheitslücken zu beobachten. Die Vermutung, dass hier gleichzeitig ein Abstieg in der Marktverbreitung vorliegt, kann nicht nachgewiesen werden, da keine belegbaren Zahlen von Betriebssystemen und deren Marktverbreitung vorliegen. Verfügbare Zahlen in diesem Bereich beziehen sich oft auf eingeschränkte Anwendungsgebiete wie Desktop-Einsatz oder in mobilen Endgeräten. Für generell einsetzbare Betriebssysteme wie NetBSD (und auch Linux) kann dies nicht pauschal verwendet werden. Gäbe es hier einen direkten Zusammenhang, dann würde die Popularität von Linux aktuell abnehmen, was nicht als gegeben angenommen werden kann.

Effektiv kann aus dem vorliegenden Daten abgeleitet werden, dass NetBSD unter allen betrachteten Systemen, die teilweise nur einzelne Anwendungsgebiete abdecken, das System ist, das mit Abstand am wenigsten Schwachstellen hat, Abbildung 4 unterstreicht dies.

Fazit

»Sicherheit« geschieht oft hinter verschlossenen Türen, was vielfach damit zusammenhängt, dass offen aufgezeigte Sicherheitsprozesse auch Schwachstellen in den Prozessen aufzeigen.

Im vorliegenden Fall wurden neben der Einsehbarkeit in den Quellcode als Produkt des NetBSD Projektes auch die Sicherheitsprozesse und die sicherheitsrelevanten Aspekte des Produkt- bzw. Softwareentwicklungsprozesses aufgezeigt. Diese können als Beispiel für ähnliche Unternehmungen dienen. Dass dabei natürlich immer an die eigenen lokalen bzw. projektspezifischen Gegebenheiten angebracht ist versteht sich von selbst.

Durch das Commitment zu offenen Prozessen im Security Bereich und ein offen einsehbares Ergebnis, das zum Peer Review bereit steht, ist hier ein starkes, vertrauenswürdiges und dadurch letztendlich sicheres Betriebssystem verfügbar.

Daraus folgt, dass »freie« (kostenlose) und »offen« (von urheberrechtlichen Auflagen weitgehend uneingeschränkte) Systeme bei Existenz entsprechender Regularien und Prozesse sehr wohl »sicher« sein können. Eben diese Prozesse beseitigen aber genannte »Sorgen« (Risiken) und Schränken dadurch auch gewisse Freiheiten ein.

In diesem Sinne gilt: Sicherheit ohne Regeln? Nein!
Aber: Sicherheit durch Transparenz? Ja!

Über den Autor

Dr. Hubert Feyrer hat an der FH Regensburg Technische Informatik studiert und an der Universität Regensburg im Fach Informationswissenschaft promoviert. Parallel war er anfangs als System- und Netzwerkadministrator sowie später als Dozent an der Hochschule Regensburg, der Uni Regensburg und am Stevens Institut Technology in Hoboken, New Jersey, USA, tätig. Nach seiner Promotion arbeitete er in der freien Wirtschaft, zuletzt als technischer Leiter bei einem Hersteller von Sicherheitslösungen mit Sitz in München wo er als technischer Leiter Teams in den Bereichen IT-Security und Datacenters leitete, Entwicklungen in den Bereichen Hard- und Software vorantrieb und die Bereiche IT-Compliance und ISMS/ISO 27001 aufbaute und betreute. Aktuell ist Dr. Feyrer als Chief Information Security Officer (CISO) und Risikomanager bei einem international agierenden Personaldienstleister der Automobilbranche tätig.

Literaturverzeichnis

Aho, A. V., Lam, M. S., Sethi, R. & Ullman, J. D. (2006). *Compilers* (2. Auflage). Upper Saddle River, NJ, USA: Prentice Hall.

Christopher Daase. (2010). Der erweiterte Sicherheitsbegriff. Zugriff am 4. Juli 2012, unter `http://www.sicherheitskultur.org/WorkingPapers/01-Daase.pdf`

Feyrer, H. (2012). Sicherheit durch Freiheit und Offenheit?: Ein Fallbeispiel. *Magdeburger Journal zur Sicherheitsforschung, 2,* 202 – 216. Zugriff am 16. Juli 2012, unter `http://www.wissens-werk.de/index.php/mjs`

Hans Halstrick and Karsten M. Decker. (2009). Informationssicherheit: Organisation vor Technik! *Organisator, 8-9.*

Hans Peter Bull. (2011). *Informationelle Selbstbestimmung - Vision oder Illusion?* (2. Auflage). Mohr Siebeck Verlag.

heiseSecurity. (2012). Flame kam angeblich als Windows-Update aufs Sys-

Year	NetBSD	FreeBSD	OpenBSD	Windows	Linux	Apple iOS
1998	2	7	3	8	14	-
1999	6	14	10	63	71	-
2000	17	55	16	55	111	-
2001	19	57	16	100	191	-
2002	17	55	15	146	171	-
2003	16	9	6	76	143	-
2004	11	29	18	128	183	1
2005	12	45	7	184	548	0
2006	35	49	28	225	516	0
2007	10	24	14	305	508	0
2008	14	35	20	226	298	0
2009	21	26	8	350	323	1
2010	4	12	1	585	311	35
2011	8	14	6	454	235	37
2012	1	2	1	122	84	67
Total:	193	433	169	3027	3707	141

Tabelle 1: Sicherheitslücken pro Jahr und Betriebssystem (1/3)

Year	Android	Mac OS X	Solaris	AIX	HP-UX	Blackberry
1998	-	1	14	8	3	-
1999	-	3	26	11	6	-
2000	-	2	8	8	17	-
2001	-	3	39	21	29	-
2002	-	13	39	23	19	-
2003	-	12	35	11	20	-
2004	-	21	30	15	14	1
2005	-	30	35	36	20	5
2006	1	88	53	27	24	2
2007	5	134	64	38	23	7
2008	2	110	74	39	16	1
2009	11	93	108	22	9	11
2010	21	212	115	15	8	11
2011	60	82	112	13	9	11
2012	94	27	27	8	5	1
Total:	194	831	779	295	222	50

Tabelle 2: Sicherheitslücken pro Jahr und Betriebssystem (2/3)

Year	Symbian	QNX	VxWorks	Simatic
1998	-	-	-	-
1999	-	-	-	-
2000	-	4	-	-
2001	-	2	-	-
2002	-	10	-	-
2003	-	0	1	-
2004	-	5	2	-
2005	1	4	5	-
2006	1	6	4	-
2007	0	0	0	-
2008	1	1	1	-
2009	2	0	0	-
2010	0	0	4	1
2011	0	2	1	1
2012	0	0	0	12
Total:	5	34	18	14

Tabelle 3: Sicherheitslücken pro Jahr und Betriebssystem (3/3)

Betriebssystem	Total
Linux	3707
Windows	3027
Mac OS X	831
Solaris	779
FreeBSD	433
AIX	295
HP-UX	222
Android	194
NetBSD	193
OpenBSD	169
Apple iOS	141
Blackberry	50
QNX	34
VxWorks	18
Simatic	14
Symbian	5

Tabelle 4: Sicherheitslücken von 1998 bis 2012

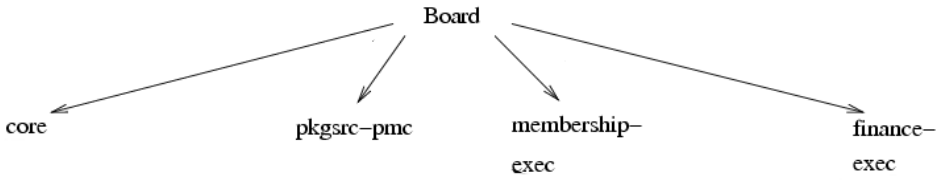

Abbildung 1: Die Organisation der NetBSD Stiftung

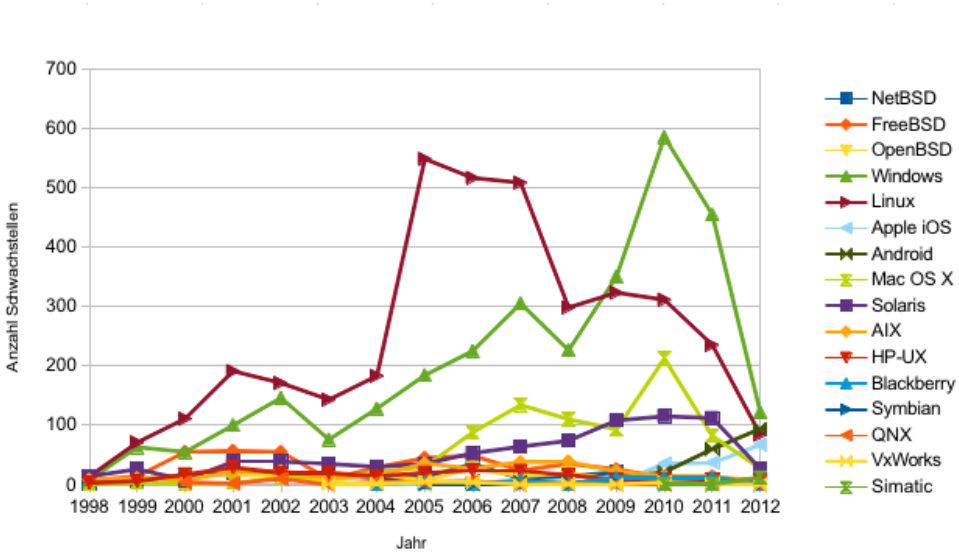

Abbildung 2: Schwachstellenfunde aller Systeme

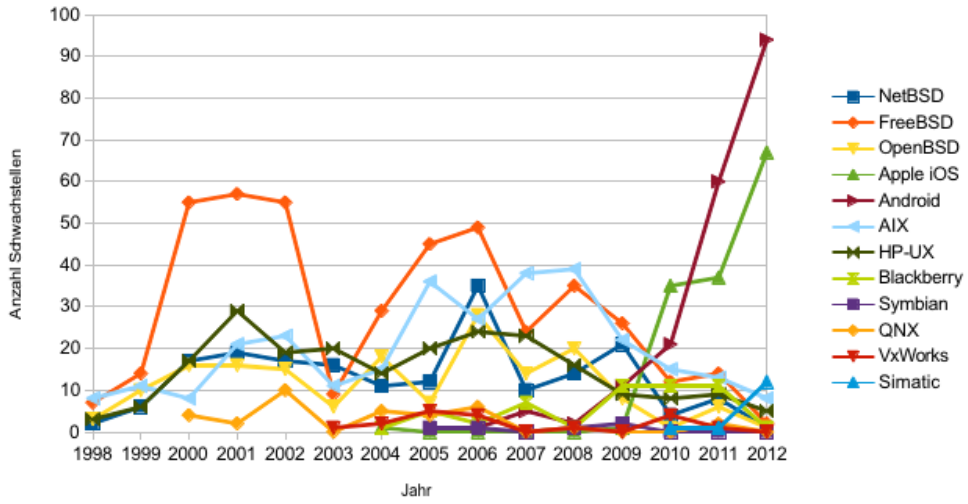

Abbildung 3: Schwachstellenfunde ohne Spitzenreiter

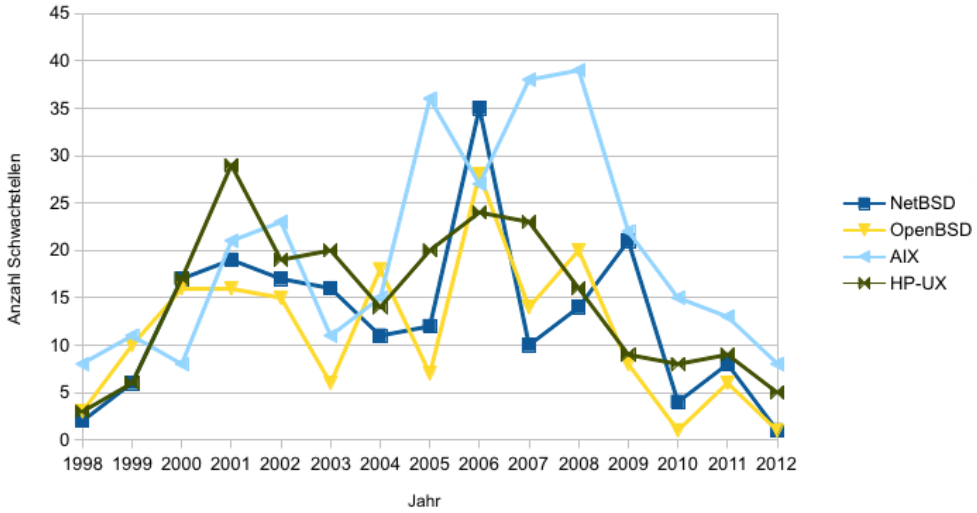

Abbildung 4: Systeme mit den wenigsten Schwachstellenfunde

tem. Zugriff am 4. Juli 2012, unter http://www.heise.de/security/meldung/Flame-kam-angeblich-als-Windows-Update-aufs-System-1603288.html

ISO/IEC. (2005). *ISO/IEC 27001 Information Security Management Systems (ISMS) standard*. Geneva: International Organization for Standardization.

Magnus Bergquist and Jan Ljungberg and Bertil Rolandsson. (2011). A Historical Account of the Value of Free and Open Source Software: From Software Commune to Commercial Commons. *Open Source Systems: Grounding Research, 365/2011*, 196–297.

Michael Schwarz and Yuri Takhteyev. (2010). Half a Century of Public Software Institutions: Open Source as a Solution to the Hold-Up Problem. *Journal of Public Economic Theory*, *12*(4), 609–639.

Mill, J. S. (2011). *Über die Freiheit* (2. Auflage). Hamburg: Felix Meiner Verlag.

Mundhenke, J. (2007). *Wettbewerbswirkungen Von Open-Source-Software und Offenen Standards Auf Softwaremärkten*. Berlin: Springer.

National Institute of Standards and Technology. (2012). National Vulnerability Database. Zugriff am 9. Juli 2012, unter http://nvd.nist.gov/

The NetBSD Foundation. (2012a). Contributing to the NetBSD project. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/contrib/

The NetBSD Foundation. (2012b). Information about NetBSD-current. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/releases/current.html

The NetBSD Foundation. (2012c). NetBSD Community. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/community/

The NetBSD Foundation. (2012d). NetBSD Exportbestimmungen. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/about/crypto-export.html

The NetBSD Foundation. (2012e). NetBSD mailing lists: tech-security. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/mailinglists/#tech-security

The NetBSD Foundation. (2012f). NetBSD Security Advisories by Date. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/support/security/advisory.html

The NetBSD Foundation. (2012g). NetBSD Security Advisories by Release. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/support/security/release.html

The NetBSD Foundation. (2012h). NetBSD: Security Issues. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/developers/security.html

The NetBSD Foundation. (2012i). New NetBSD developers application procedure. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/foundation/policies/application-procedure.html

The NetBSD Foundation. (2012j). PGP Key Management Guide for NetBSD developers. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/developers/pgp.html

The NetBSD Foundation. (2012k). Security and NetBSD. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/support/security/

The NetBSD Foundation. (2012l). The NetBSD Foundation. Zugriff am 4. Juli 2012, unter http://www.netbsd.org/foundation/

The NetBSD Foundation. (2012m). The NetBSD Project. Zugriff am 4. Juli 2012, unter http://www.NetBSD.org/

Urs Fässler. (2011). *Freie Software*. ETH Zürich. Zugriff am 4. Juli 2012, unter http://n.ethz.ch/~ursf/download/freiesoftware.pdf

Wikipedia. (2012). Sicherheit — Wikipedia, Die freie Enzyklopädie. Zugriff am 4. Juli 2012, unter http://de.wikipedia.org/w/index.php?title=Sicherheit&oldid=102196504