Sicherheit von Open Source Software

Wie sicher ist Open Source Software?

Lukas Kairies

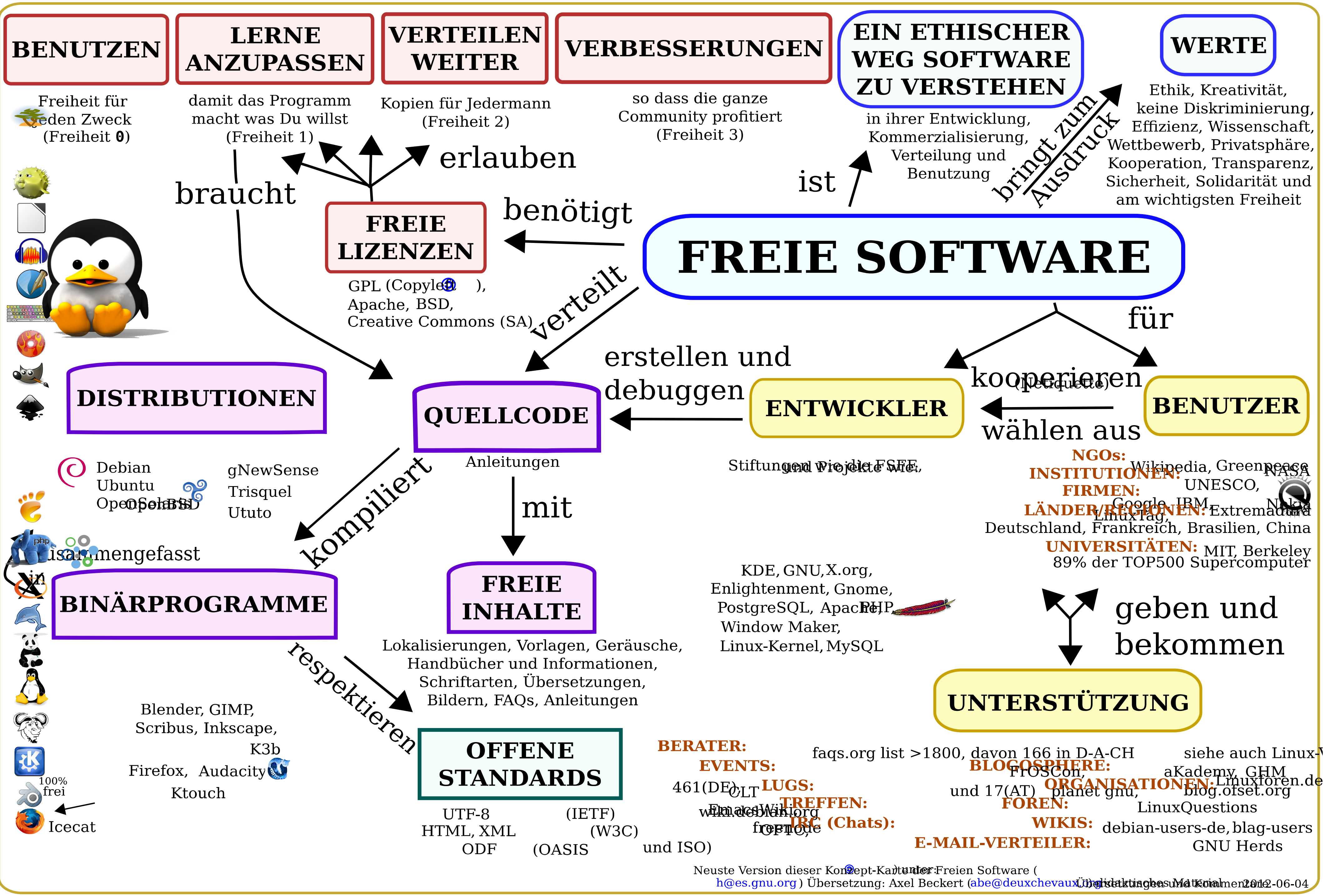
Gliederung

1. Begriffseinführung
    1. Freie Software
    2. Open Source Software
2. Sicherheitsphilosophien
    1. Open Source Software
    2. proprietäre Software
3. Angriffsmöglichkeiten
4. Fallbeispiel: Heartbleed

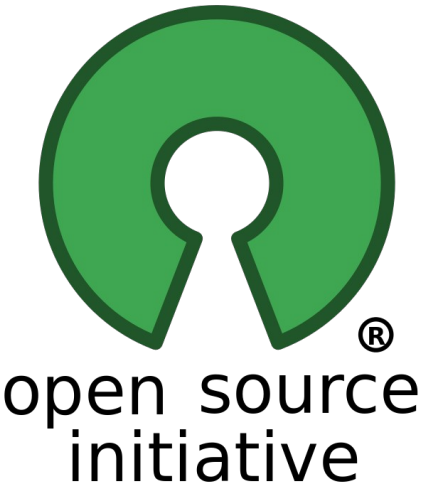
Freie Software

- Free Software Foundation (Richard Stallmann 1985)
  - Freiheit 0: Das Programm zu jedem Zweck auszuführen.
  - Freiheit 1: Das Programm zu untersuchen und zu verändern.
  - Freiheit 2: Das Programm zu verbreiten.
  - Freiheit 3: Das Programm zu verbessern und diese Verbesserungen zu verbreiten, um damit einen Nutzen für die Gemeinschaft zu erzeugen
- „free speech not free beer“
- Sicht auf Anwender und Gesellschaft

FREE SOFTWARE FOUNDATION


BENUTZEN
Freiheit für jeden Zweck (Freiheit 0)
LERNE ANZUPASSEN
damit das Programm macht was Du willst (Freiheit 1)
VERTEILEN WEITER
Kopien für Jedermann (Freiheit 2)
VERBESSERUNGEN
so dass die ganze Community profitiert (Freiheit 3)
EIN ETHISCHER WEG SOFTWARE ZU VERSTEHEN
in ihrer Entwicklung, Kommerzialisierung, Verteilung und Benutzung
WERTE
Ethik, Kreativität, keine Diskriminierung, Effizienz, Wissenschaft, Wettbewerb, Privatsphäre, Kooperation, Transparenz, Sicherheit, Solidarität und am wichtigsten Freiheit
erlauben
braucht
FREIE LIZENZEN
GPL (Copyleft), Apache, BSD, Creative Commons (SA)
benötigt
ist
bringt zum Ausdruck
FREIE SOFTWARE
verteilt
für
erstellen und debuggen
kooperieren (Netiquette)
wählen aus
ENTWICKLER
BENUTZER
DISTRIBUTIONEN
QUELLCODE
Anleitungen
Debian
Ubuntu
OpenSolaris
OpenBSD
gNewSense
Trisquel
Ututo
Stiftungen wie die FSFE,
und Projekte wie:
NGOs: Wikipedia, Greenpeace
INSTITUTIONEN: UNESCO, NASA
FIRMEN: Google, IBM, Nokia
LÄNDER/REGIONEN: Extremadura
LinuxTag,
Deutschland, Frankreich, Brasilien, China
UNIVERSITÄTEN: MIT, Berkeley
89% der TOP500 Supercomputer
kompiliert
mit
zusammengefasst in
BINÄRPROGRAMME
FREIE INHALTE
Lokalisierungen, Vorlagen, Geräusche, Handbücher und Informationen, Schriftarten, Übersetzungen, Bildern, FAQs, Anleitungen
KDE, GNU, X.org, Enlightenment, Gnome, PostgreSQL, Apache, PHP, Window Maker, Linux-Kernel, MySQL
geben und bekommen
UNTERSTÜTZUNG
respektieren
Blender, GIMP, Scribus, Inkscape, K3b
Firefox, Audacity
Ktouch
100% frei
Icecat
OFFENE STANDARDS
UTF-8 (IETF)
HTML, XML (W3C)
ODF (OASIS und ISO)
BERATER: faqs.org list >1800, davon 166 in D-A-CH
EVENTS: FrOSCon, aKademy, GHM
LUGS: 461(DE), 17(AT)
CLT
TREFFEN:
IRC (Chats): freenode, OFTC
BLOGOSPHERE: planet gnu, blog.ofset.org
ORGANISATIONEN:
FOREN: linuxforen.de, LinuxQuestions
WIKIS: EmacsWiki, wiki.debian.org
E-MAIL-VERTEILER: debian-users-de, blag-users
GNU Herds
siehe auch Linux-
Neuste Version dieser Konzept-Karte der Freien Software (h@es.gnu.org) Übersetzung: Axel Beckert (abe@deuxchevaux.org) didaktisches Material 2012-06-04

Open Source Software



- Open Source Initiative
  - 1998 Bruce Perens und Eric S. Raymond
- Definition im wesentlichen gleich mit Free Software
- Betonung auf Überlegenheit des Entwicklungsprozesses
- Abgrenzung von Free Software zur einfacheren Vermarktung

Gliederung

1. Begriffseinführung
    1. Freie Software
    2. Open Source Software
2. Sicherheitsphilosophien
    1. Open Source Software
    2. proprietäre Software
3. Angriffsmöglichkeiten
4. Fallbeispiel: Heartbleed

Sicherheitsphilosophie – Open Source Software

- „Open Source Software ist sicherer“
- Jeder kann Quellcode einsehen
  - Experten und Anfänger gleicher maßen
- Öffentlicher Reviewprozess:
  1. Änderung wird eingereicht
  2. Änderung werden über Mailingliste diskutiert und ggf. angepasst
  3. Änderung wird eingepflegt oder abgelehnt
- Annahme: Mehr Code reviews durch „in-house“ reviews und Außenstehende (Viele-Augen-Prinzip)
  - Besonders gegeben bei kommerziell genutzten Open Source Projekten

Sicherheitsphilosophie – proprietäre Software

- „ Security Through Obscurity“
- Nichtverfügbarkeit von Quellcode erschwert/verzögert finden von Schwächen für Angreifer
  - U.a. „Reverse Engineering“ nötig
- Keine Vorteile durch veröffentlichen des Quellcodes
- Wahren von Geschäftsgeheimnissen

Gliederung

1. Begriffseinführung
    1. Freie Software
    2. Open Source Software
2. Sicherheitsphilosophien
    1. Open Source Software
    2. proprietäre Software
3. Angriffsmöglichkeiten
4. Fallbeispiel: Heartbleed

Angriffsmöglichkeiten - Quellabängig

- Bufferoverflow/SQL Injection
- Patch Reverse Engineering (day zero attack)
  - Rückführen von Sicherheitslücken aus Patches
  - Angriff auf ungepatchte Systeme
  - Zeitkritisch
  - Tools zum automatischen Erzeugen von Exploits
  - Betrifft Open und Closed Source

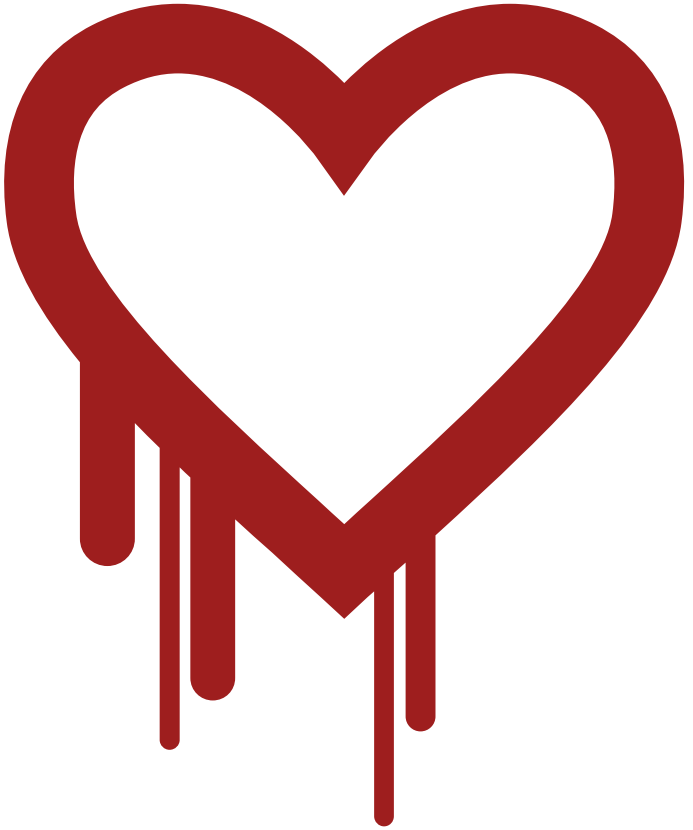
Angriffsmöglichkeiten - Quellunabhängig

- Angriff mit Nutzerbeteiligung
- Brute Force Angriffe
- Protokollschwachstellen
- Insiderjobs

Gliederung

1. Begriffseinführung
    1. Freie Software
    2. Open Source Software
2. Sicherheitsphilosophien
    1. Open Source Software
    2. proprietäre Software
3. Angriffsmöglichkeiten
4. Fallbeispiel: Heartbleed

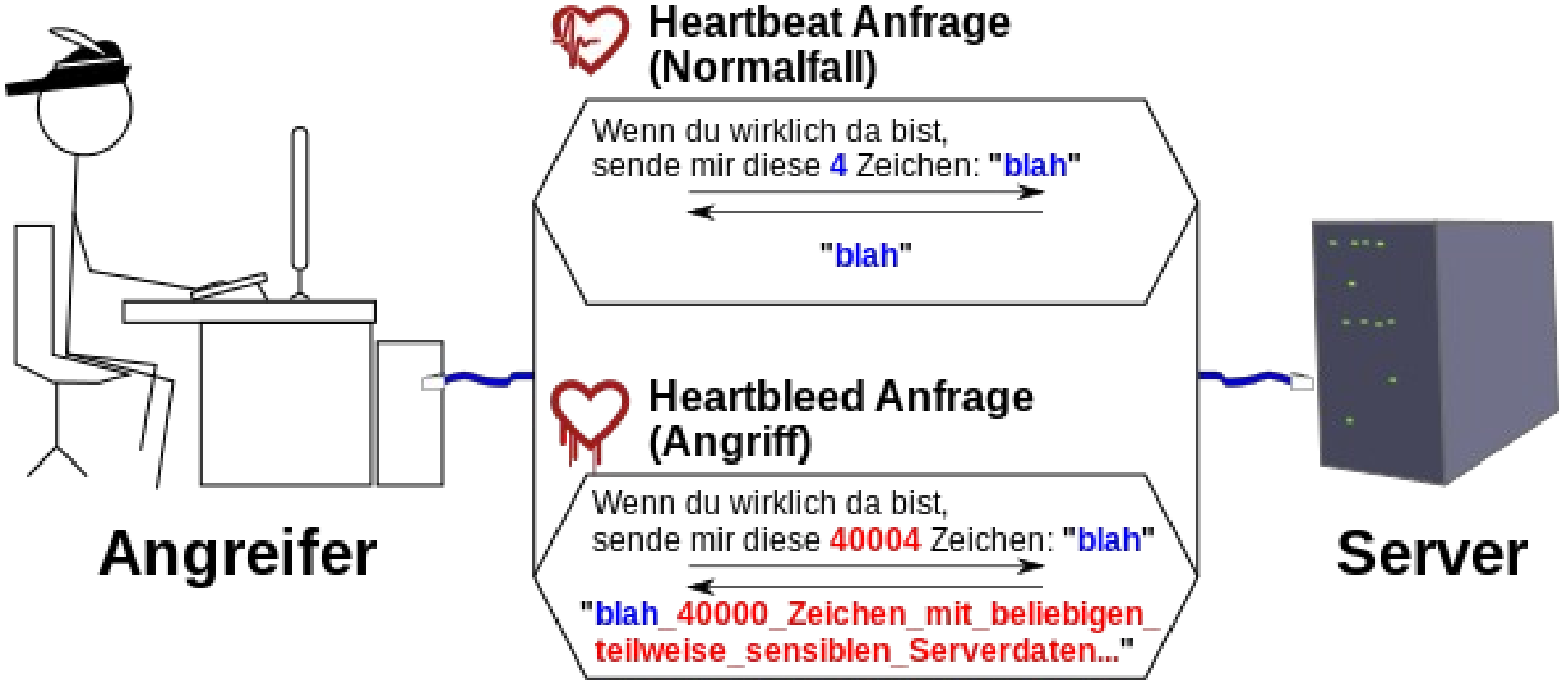
4. Heartbleed Bug



Bruce Schneier: “Catastrophic is the right word. On the scale of 1 to 10, this is an 11.”

Heartbleed Bug - Funktionsweise

- Abgreifen von beliebigen Serverdaten aus Arbeitsspeicher: Passwörter, Private Schlüssel, Session Cookies,...



Heartbleed Bug - Entstehung

- Am 31.Dezember 2011 in Codebasis eingepflegt und am 14.März 2012 veröffentlicht
- Code entstand im Rahmen einer Dissertation eines Studenten
- Entdeckung am 7. April 2014
  - Bug bestand für 27 Monate

Heartbleed Bug – Folgen

- 24-55% aller Webseiten (HTTPS) potentiell betroffen
  - mindestens 44 der Top 100 Websites betroffen
- Grundsätzlich müssen alle betroffenen Server als kompromittiert angesehen werden
- Massenweiser Widerruf von Zertifikaten nötig
  - Keine ausreichende Infrastruktur vorhanden
- Auch betroffen: VoIP-Telefonie, Netzwerkdrucker, Router,...
- Kanada: 19-Jähriger konnte 900 Sozialversicherungsnummern von Servern des Finzamtes entwenden

Heartbleed – Folgen für OpenSSL

- Theo de Raadt (OpenBSD): Sicherheitsmechanismen sind zugunsten der Leistung umgangen worden
  - Fork LibreSSL zur Codebereinigung
- Quellcode wurde intensiv untersucht
  - Weitere Lücken wurden geschlossen
- Forderung nach mehr finanzieller Unterstützung

Quellen

- http://de.wikipedia.org/wiki/Open_Source
- http://de.wikipedia.org/wiki/Open_Source_Initiative#Definition_von_Open_Source
- http://de.wikipedia.org/wiki/Free_Software_Foundation
- http://de.wikipedia.org/wiki/Freie_Software
- http://de.wikipedia.org/wiki/Heartbleed
- media.ccc.de
- Clarke, Russell, David Dorwin, and Rob Nash. "Is Open Source Software More Secure?." (1999).