Open Source und Datensicherheit - ein Widerspruch?

Hermann Gottschalk
Thomas Birnthaler

eMail: info@ostc.de
Web: http://www.ostc.de

Inhaltsverzeichnis

1 Was ist (Daten)Sicherheit?

- **Ziele von Datensicherheit**
  - Verfügbarkeit
  - Integrität
  - Vertraulichkeit
  - Decodierbarkeit
  - Transferierbarkeit

Absolute (Daten)Sicherheit ist nicht erreichbar!

- **Maßnahmen**
  - Physische / räumliche Sicherung
  - Zugriffskontrolle
  - Fehlertoleranz
  - Kryptographie
  - Standards

1 Was ist Open Source?

- **Open Source Software** (oder auch Free Software) genügt folgenden **10 Regeln**:
  - (Kosten)freie Weitergabe des Programms
  - Quellcode frei verfügbar
  - Veränderungen / Ableitungen zulässig
  - Veränderungen erkennbar
  - Keine Diskriminierung von Personen / Gruppen
  - Keine Beschränkung des Einsatzfeldes
  - Software-Weitergabe = Lizenz-Weitergabe
  - Keine Beschränkung auf Produkt-Kombinationen
  - Uneingeschränkte Weitergabe mit anderer Software
  - Technologie-neutral

1 Was ist Open Source?

- **Unterschiedliche Lizenz-"Spielarten"**
  - GNU General Public License (GPL)
  - GNU Library or "Lesser" Public License (LGPL)
  - BSD License
  - MIT License
  - IBM Public License
  - W3C License

- Software unter einer dieser Lizenzen erfüllt die 10 Open Source Regeln

- **Copyright** an der Software bleibt uneingeschränkt beim Urheber

1 Was ist Linux?

- **Linux** = vollständige Neuprogrammierung von UNIX
  - **Beginn**: 1990
  - **Autor**: Finnischer Student Linus Torvalds
  - **Besonderheit**: Sofortige Freigabe unter der GPL
  - **Basis**: GNU-Projekt der Free Software Foundation (steht ebenfalls unter der GPL)
  - **UNIX-Standards** wie SVID und POSIX erfüllt

- **Das Paradebeispiel** für Open Source Software

- Ohne offene UNIX-Standards wäre
  - dieses Projekt kaum möglich gewesen bzw.
  - hätte wesentlich länger gedauert

2 Bekannte Beispiele für Open Source

- Open Source ist **kein neuer Gedanke**:
  - **Kryptographie**
    - Security by obscurity doesn't work! (Bruce Schneier)
    - Klassisches Beispiel "**Enigma**"
      - Konstruktion nicht offengelegt
      - "Knackbar" aufgrund Kombination von
        - Systematischen Schwächen
        - Anwendungsfehlern
  - **Firma GRUNDIG**
    - Radio-Bausatz "Heinzelmann"
    - Schaltpläne zu jedem Gerät
  - **Internet**
    - Welche Technik war je erfolgreicher?

3 Anforderungen an zentrale Software

- **Zu zentraler Software zählen**
  - Programmiersprachen
  - Betriebssysteme
  - Firewalls / Virenscanner
  - Mail-Server
  - DNS-Server
  - Web-Server
  - Internet-Browser
  - Grafische Oberflächen
  - Office-Pakete

- Es gibt ausgereifte Open Source Software für diese Zwecke!

3 Anforderungen an zentrale Software

- **Anforderungen an zentrale Software**
  - Erfüllt Standards
  - Funktionalität verifizierbar
  - Sicher
  - Gut dokumentiert
  - Möglichst fehlerfrei
  - Herstellerunabhängig
  - Plattformunabhängig

- Open Source Software erfüllt diese Anforderungen perfekt!

4 Open Source und Sicherheit

Pro: Sicherheit durch offene und geteilte Entwicklung

- Viele unabhängige Entwickler / Tester / Anwender
  - Prüfen sich gegenseitig ("Peer Review")
  - Teilen sich die Last des Testens
- Gesamter Erstellungsprozess sichtbar
  - Probleme gut erkennbar und korrigierbar
- Öffentlicher Code verpflichtet zu
  - Offenheit
  - Qualität
  - Dokumentation

4 Open Source und Sicherheit

Pro: Schnelles Beheben von Fehlern

- Fehler sind bei komplexer Software nicht ausschließbar
- Quellcode verfügbar und änderbar
  - Fehler gut kommunizierbar
  - Fehler gut suchbar
  - Fehler von jedem behebbar (im Prinzip)
- Entwicklungshistorie verfügbar
  - Nachvollziebar, wann von wem welcher Fehler behoben wurde

4 Open Source und Sicherheit

Pro: Volle Kontrolle der Funktionalität

- Quellcode verfügbar
  - Funktionalität sichtbar
  - Eigene Version selbst herstellbar
  - Unerwünschte Funktionalität weglassbar
- Beiträge anderer Entwickler / Anwender
  - Leicht integrierbar
  - Expertise und Ressourcen anderer nutzen

4 Open Source und Sicherheit

Pro: Enthält nur wirklich nötige Funktionalität

- Beruht auf dem unmittelbaren Anwenderbedarf
  - Nur nützliche Funktionalität enthalten
  - Sparsamer Ressourcenverbrauch
  - Hohe Zuverlässigkeit
  - Benutzerfreundliche Anwendungen
- Zusätzlich benötigte Funktionalität leicht ergänzbar

4 Open Source und Sicherheit

Pro: Kontinuierliche Anpassung statt teurer Releases

- Weiterentwicklung in kleinen Schritten
  - Änderungen überschaubar
  - Wahl ob mitmachen oder nicht
  - "Jetzt-oder-Nie"-Problematik gering
- Alte Versionen werden weiterhin gepflegt
  - Solange Bedarf besteht
  - Im Extremfall selbst übernehmen

4 Open Source und Sicherheit

Pro: Unabhängigkeit von Anbietern

- Viele unabhängige Support-Anbieter
  - Freie Wahl
  - Wechsel vom ursprünglichen Anbieter möglich
  - Bedarf anpassbar an eigenes Know-How

- Quellcode vorhanden
  - Im Notfall selber eingreifen oder
  - Externen Dienstleister eingreifen lassen

Pro: Erfüllt offene Standards

- Austauschbarkeit
  - der Hardware-Plattform
  - der Betriebssystem-Umgebung
  - der Software-Komponenten
- Unabhängigkeit von
  - Herstellern / Lieferanten
  - Hardware
- Bei Bedarf schrittweiser Wechsel möglich

4 Open Source und Sicherheit

Pro: Effiziente Ressourcen-Nutzung

- Anwenderbedarf entscheidet über Funktionalitäten
  - Schlanke Software
  - Geringer Hardware-Anforderungen
- Hard- und Softwareneutralität
  - Zwingt zu klaren Konzepten
- Einhaltung von Standards
  - Erlaubt Wiederverwendung von Komponenten

4 Open Source und Sicherheit

Pro: Fördert

- Wettbewerb
- Standards
- Dokumentation
- Know-How-Transfer
- Lernen
- Unterrichten
- Freie Produktwahl

4 Open Source und Sicherheit

Pro: Günstige Anschaffung und TCO (Total Cost of Ownership)

- Open Source Lizenz
  - Erlaubt nur geringfügige Kaufkosten
  - Einsatz auf beliebig vielen Systemen
  - Keine Kosten bei Releasewechsel

- Gespartes Geld investieren in
  - Anwenderschulung
  - Weiterentwicklung
  - Sicherheitstechnik

- Häufigstes Argument, aber eigentlich wenig relevant.

Kontra: Software-Entwicklung

- Kein **rechtlich Verantwortlicher**
  - Haftung gibt es bei normaler Lizenz-Software auch nicht
- **Entwicklungsrichtung** ist nicht festlegt.
  - Von Anwendern / Entwicklern gemeinsam entschieden
- **Pflege der Software** nicht gesichert
  - Bei Firmenaufkauf / -konkurs ebenfalls nicht
  - Im Prinzip kann man dies selber übernehmen
- **Know-How** ist nicht geschützt
  - Man kann daraus lernen
  - Urheberrecht / Copyright bleibt uneingeschränkt gültig

4 Open Source und Sicherheit

Kontra: Kommerzielle Nutzung

- Man kann kein **Geld** damit verdienen
  - Mit Lizenzverkauf in der Tat nicht
  - Mit Support / Dienstleistung sehr wohl

- Kein **Marketing**, das das Produkt vorantreibt
  - Führt häufig zu "Featuritis" (ständiger Releaswechsel)
  - Erhöht die Kosten

- **Kommerzielle Unterstützung** schlecht
  - Dienstleister sind an zufriedenen Kunden interessiert, da sie sich nicht über das Lizenzgeschäft interessieren
  - Das regelt der Markt (IBM, HP, SUN, Oracle, SAP, ...)

4 Open Source und Sicherheit

Kontra: Sicherheit

- Open Source liefert **nicht automatisch Sicherheit und Qualität**
  - Gilt für viele andere Entwicklungs-Methoden auch
- **Fehler** in Open Source Programme liegen offen zu Tage
  - Von jedermann sofort ausnutzbar
- Weltweit verteilte Entwickler sind **nicht kontrollierbar**
  - Einbau von "subversiven" Funktionalitäten
- Kurzfristig stimmt das, langfristig gilt aber immer noch:
  "**Security by obscurity doesn't work**" (Bruce Schneier)

Quellen zum Thema Open Source

- **Open Source** (www.opensource.org)
- **Open Source Deutschland** (www.opensource.co.at)
- **Free Software Foundation** (www.fsf.org)
- **Free Software Foundation Europe** (www.fsf-europe.org)
- **Linux-Kern** (www.kernel.org)
- **GNU** (www.gnu.org)
- **FreeBSD** (www.freebsd.org)
- **NetBSD** (www.netbsd.org)
- **OpenBSD** (www.openbsd.org)
- **Homepage von Richard M. Stallman** (www.stallman.org)
- **Homepage von Eric S. Raymond** (catb.org/~esr)

Vielen Dank für Ihre Aufmerksamkeit!
Für Fragen stehen wir Ihnen zur Verfügung

Hermann Gottschalk
Thomas Birnthaler

eMail: info@ostc.de
Web: http://www.ostc.de