Unsichere Software - Eine systemische Betrachtung

Robert A. Gehring

TU Berlin

(rag@cs.tu-berlin.de)

These:
Die Unsicherheit von Software hat

- Technische Ursachen
- Rechtliche Ursachen
- Ökonomische Ursachen

Beispiele aus der Wirklichkeit

- 1998: Der durch Viren verursachte Schaden beträgt ca. 6 Milliarden US$
- 1999: "I love you"-Virus verursacht Schaden von ca. 12 Milliarden US$
- 2000: "I love you"-Virus verursacht innerhalb von zwei Wochen einen Schaden von ca. 6,7 Milliarden US$

(Figures from McAfee 2001)

Technische Ursachen (I)

- **Unvollständige Spezifikationen:**

  «[M]odern Systems have so many components and connections-some of them not even known by the systems' designers, implementers, or users-that insecurities always remain.»

  (Schneier 2000: xii)

Technische Ursachen (II)

- **Unvollständige Tests:**

  «The developers are so in tune with what [the system] should do, they cannot see what it might be able to do.»

  (Pipkin 2000: 75)

Technische Ursachen (III)

- **Unvollständige Tests:**

  Ein Test kann nur die Existenz eines Fehlers belegen.

  Ein Test kann nicht belegen, daß keine Fehler vorhanden sind.

  (Floyd 1997: 664; Kaner 1997a)

Ökonomische Ursachen (I)

- **Proprietäre Software ist ein Produkt das vermarktet werden *muß*.**
- **Anbieter und Käufer von Software verfügen über stark asymmetrische Informationen** (a "market for lemons", Akerlof 1970)**.**
- **Netzwerk-Externalitäten** (Shapiro/Varian 1999) **haben einen starken -schlechten- Einfluß auf die Produktqualität.**

Ökonomische Ursachen (II)

- **Die Kosten für Testen und Fehlerbeseitigung zu minimieren ist ökonomisch *rationales Verhalten*.**
- **Die Service-Kapazitäten der Hersteller von Standard-Software sind begrenzt.**
- **Anbieter von Individualsoftware verdienen an: *«maintenance ... successive updates, patches and versions»*** (Levinson 2001)**.**

Rechtliche Ursachen

- Es gibt *keine wirksamen Haftungsregelungen*, die auf den Vertrieb unsicherer Standard-Software anwendbar sind.
- Copyright und Urheberrecht berücksichtigen Sicherheitsanforderungen *nicht*.
- Patentschutz kann sicherere Konkurrenzprodukte blockieren.

Schwachstellen im Urheberrecht

- **Reverse engineering weitgehend verboten.**
- **Reparatur von Software ist verboten.**
- **Sicherheitserweiterung sind verboten.**
- **Keine Haftung für *Sprachwerke.***
- **Software wird nicht als *«a machine»* betrachtet** (Samuelson et. al. 1994).

Schwachstellen des Patentrechts

- Binärvertrieb wird gefördert, Qualitätstransparenz behindert.
- Kompatible Produkte mit besserer Qualität und Sicherheit können blockiert werden.
- Sicherheitstechnologie kann patentiert werden, was die schnelle Verbreitung behindert.
- Verfahren zur Absicherung können als «*business model patent*» geschützt werden.

Wie läßt sich die IT-Sicherheit verbessern?

- Wir brauchen eine adäquate *Risikomanagement-Strategie*.

Das *Open Source Software*-Modell für die Entwicklung und den Vertrieb von Software könnte der Grundstein einer solchen Risikomanagement-Strategie sein.

Die Stärken des OSS-Ansatzes

- *Unabhängige* Code-Kontrolle wird ermöglicht.
- Berücksichtigt beim Copyright-Schutz die Spezifika von Software (gewährt z.B. das Recht zur Modifikation).
- Weist kurze Reaktionszeiten bei sicherheitsrelevanten Vorfällen auf.
- Macht Qualität transparent (durch Quellcode-Vertrieb).

Die Gefährdung durch Patente

- Quellcode kann leicht hinsichtlich Patentverletzungen geprüft werden - Binärcode nicht (u.a. wg. "reverse engineering"-Verbot im Urheberrecht).
- Gründliche Patentrecherche erfordert die Hilfe von Patentanwälten. Unabhängige Entwickler können sich das nicht leisten.
- Unabhängige Entwickler, KMUs können sich die Verteidigung gegen eine (ggf. nicht berechtigte) Patentverletzungsklage nicht leisten.

Empfehlung

« **PP-1:** Der Umgang mit dem Quelltext von Computerprogrammen muss patentrechtlich privilegiert werden. Das Herstellen, Anbieten, in Verkehr bringen, Besitzen oder Einführen des Quelltextes eines Computerprogrammes in seiner jeweiligen Ausdrucksform muss vom Patentschutz ausgenommen werden.»

(Lutterbeck/Horns/Gehring 2000)

Referenz

Dipl.-Inform. Robert A. Gehring
TU Berlin
Informatik und Gesellschaft

http://ig.cs.tu-berlin.de/ap/rg/index.html