



Open Source und Sicherheit

Matthias Bauer

`matthiasb@acm.org`

OpenSource != Quellen lesbar Gegenbeispiel: Solaris, NFR, Hushmail,...

Diverse Studien behaupten diverses über die Sicherheit von geschlossenen/offenen Systemen. Ist methodisch schwer zu überschauen (linux-distribs == Linux? Menge **aller** sicherheitsrelevanter Software? oder nur die im Einsatz? Bezifferung der Schäden?)

Geschlossene Implementationen/Verfahren bringen zumindest in Software nicht den Vorteil der Geheimhaltung (kann Disassembliert werden).

Sicherheit ist schwer

- Schwer zu definieren:

Sicherheit ist schwer

- Schwer zu definieren:
 - Sicher gegen welchen Angreifer?

Sicherheit ist schwer

- Schwer zu definieren:
 - Sicher gegen welchen Angreifer?
 - Sicher unter welchen Annahmen?

Sicherheit ist schwer

- Schwer zu definieren:
 - Sicher gegen welchen Angreifer?
 - Sicher unter welchen Annahmen?
- Schwer zu kriegen:

Sicherheit ist schwer

- Schwer zu definieren:
 - Sicher gegen welchen Angreifer?
 - Sicher unter welchen Annahmen?
- Schwer zu kriegen:
 - Kryptographischer Verfahren werden manchmal gebrochen.

Sicherheit ist schwer

- Schwer zu definieren:
 - Sicher gegen welchen Angreifer?
 - Sicher unter welchen Annahmen?
- Schwer zu kriegen:
 - Kryptographischer Verfahren werden manchmal gebrochen.
 - Wenn das Design korrekt ist, kann die Implementation immer noch kritische Fehler enthalten.

Sicherheit ist schwer – cont.

- Die Sicherheit einer Applikation hängt immer von der Sicherheit des Betriebssystems und meist auch von der vieler anderer Komponenten ab.

Offenheit als Sicherheitsvorkehrung

- Bereits im 19. Jahrhundert formuliert (Kerkhoffs):

Offenheit als Sicherheitsvorkehrung

- Bereits im 19. Jahrhundert formuliert (Kerkhoffs):

Die Sicherheit sollte nicht von der Geheimhaltung des Verfahrens abhängen, sondern von Schlüsseln, die gewechselt werden können.

Offenheit als Sicherheitsvorkehrung

- Bereits im 19. Jahrhundert formuliert (Kerchoffs):

Die Sicherheit sollte nicht von der Geheimhaltung des Verfahrens abhängen, sondern von Schlüsseln, die gewechselt werden können.

- Beispiele für Schlüssel: EC-Card PINs, Passwörter, Chipkarten.

Offenheit – cont.

- Das Design praktisch aller modernen Verschlüsselungsverfahren liegt offen: Advanced Encryption Standard (AES), Secure Socket Layer (SSL, https), IPsec, ...

Offenheit – cont.

- Das Design praktisch aller modernen Verschlüsselungsverfahren liegt offen: Advanced Encryption Standard (AES), Secure Socket Layer (SSL, https), IPsec, ...
- Idee: Viele Augen können Fehler schnell finden.

Mal wirtschaftlich

- *Sales value vs. Use value.*

Mal wirtschaftlich

- *Sales value vs. Use value.*
- Wenn es keine Produkthaftung gibt, hat der Softwarehersteller keine Motivation, Software sicher und verlässlich zu machen. Denn:

Mal wirtschaftlich

- *Sales value vs. Use value.*
- Wenn es keine Produkthaftung gibt, hat der Softwarehersteller keine Motivation, Software sicher und verlässlich zu machen. Denn:
- First mover gets the market.

10. DISCLAIMER OF WARRANTIES.

10.1 TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE LICENSER AND ITS SUPPLIERS PROVIDE THE PRODUCT, AND ANY (IF ANY) SUPPORT SERVICES RELATED TO THE PRODUCT ("SUPPORT SERVICES") AS IS AND WITH ALL FAULTS; AND THE LICENSER AND ITS SUPPLIERS HEREBY DISCLAIM WITH RESPECT TO THE PRODUCT AND SUPPORT SERVICES ALL WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY (IF ANY) WARRANTIES OR CONDITIONS OF OR RELATED TO: TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, LACK OF VIRUSES, ACCURACY OR COMPLETENESS OF RESPONSES, RESULTS, LACK OF NEGLIGENCE OR LACK OF WORKMANLIKE EFFORT, QUIET ENJOYMENT, QUIET POSSESSION, AND CORRESPONDENCE TO DESCRIPTION. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE PRODUCT AND ANY SUPPORT SERVICES REMAINS WITH THE LICENSEE.

...

12.1 NOT WITHSTANDING ANY DAMAGES THAT YOU MIGHT INCUR FOR ANY REASON WHATSOEVER (INCLUDING, WITHOUT LIMITATION, ALL DAMAGES REFERENCED ABOVE AND ALL DIRECT OR GENERAL DAMAGES), THE ENTIRE LIABILITY OF THE LICENSER AND ANY OF ITS SUPPLIERS UNDER ANY PROVISION OF THIS EULA AND YOUR EXCLUSIVE REMEDY FOR ALL OF THE FOREGOING SHALL BE LIMITED TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE PRODUCT OR U.S.\$5.00. THE FOREGOING LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.

Mal wirtschaftlich

Instant Indemnity

- Amerikanische Lizenzen schliessen Produkthaftung komplett aus (Disclaimer of Warranties).

Mal wirtschaftlich

Instant Indemnity

- Amerikanische Lizenzen schliessen Produkthaftung komplett aus (Disclaimer of Warranties).
- In Deutschland ist Produkthaftung einklagbar, endet praktisch immer in Vergleichen, um Präzedenzfälle zu vermeiden.

Mal wirtschaftlich

Instant Indemnity

- Amerikanische Lizenzen schliessen Produkthaftung komplett aus (Disclaimer of Warranties).
- In Deutschland ist Produkthaftung einklagbar, endet praktisch immer in Vergleichen, um Präzedenzfälle zu vermeiden.
- Jeder Kunde klagt also allein (Zivilklage).

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG
- SELinux (NSA)

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG
- SELinux (NSA)
- stunnel/vtun

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG
- SELinux (NSA)
- stunnel/vtun
- NFR (kein OS)

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG
- SELinux (NSA)
- stunnel/vtun
- NFR (kein OS)
- snort

Erfolgreiche offene Security Software

- OpenBSD (DoD, NSA, NASA, Genua, Pipenet)
- OpenSSH
- GnuPG
- SELinux (NSA)
- stunnel/vtun
- NFR (kein OS)
- snort
- OpenSSL (Bibliothek, z.B. in Mozilla)

Vorteile offener Sicherheits-Software

- Peer-Review bei Design möglich (leider selten der Fall).

Vorteile offener Sicherheits-Software

- Peer-Review bei Design möglich (leider selten der Fall).
- Review der Implementation hilft Fehler vermeiden.

Vorteile offener Sicherheits-Software

- Peer-Review bei Design möglich (leider selten der Fall).
- Review der Implementation hilft Fehler vermeiden.
- Falls Fehler entdeckt werden, kann sie potentiell jede/r beheben. (nur unter Open Source Lizenzen).

Vorteile offener Sicherheits-Software

- Peer-Review bei Design möglich (leider selten der Fall).
- Review der Implementation hilft Fehler vermeiden.
- Falls Fehler entdeckt werden, kann sie potentiell jede/r beheben. (nur unter Open Source Lizenzen).
- “Window of Exposure” wird kürzer.

Vorteile – cont.

- Gute Implementierungen können in anderen Projekten eingebunden werden (nicht unproblematisch). Da nicht jeder das Rad neu erfinden muss, benutzen viele (hoffentlich) gute, vorhandene Räder.

Vorteile – cont.

- Gute Implementierungen können in anderen Projekten eingebunden werden (nicht unproblematisch). Da nicht jeder das Rad neu erfinden muss, benutzen viele (hoffentlich) gute, vorhandene Räder.
- Interoperabilität ist bei offenen Quellen leichter zu erreichen. Das ist im Netzwerkbereich essentiell.

Vorteile – cont.

- Gute Implementationen können in anderen Projekten eingebunden werden (nicht unproblematisch). Da nicht jeder das Rad neu erfinden muss, benutzen viele (hoffentlich) gute, vorhandene Räder.
- Interoperabilität ist bei offenen Quellen leichter zu erreichen. Das ist im Netzwerkbereich essentiell.
- Der Erfahrung nach propagieren Bugfixes schneller, weil man nicht die alte Version kaufen+installieren muss, um danach alle bisherigen Fixes draufzupatchen.

Keine Garantie für Qualität

- Offener Code ist nicht notwendig auch lesbar (z.B. PGP von Zimmermann und später NAI, Inc.).

Keine Garantie für Qualität

- Offener Code ist nicht notwendig auch lesbar (z.B. PGP von Zimmermann und später NAI, Inc.).
- Den vielen Augen stehen tausende von Projekten gegenüber.

Keine Garantie für Qualität

- Offener Code ist nicht notwendig auch lesbar (z.B. PGP von Zimmermann und später NAI, Inc.).
- Den vielen Augen stehen tausende von Projekten gegenüber.
- Fehler im Design werden auf Code–Ebene nicht erkannt.

Keine Garantie für Qualität

- Offener Code ist nicht notwendig auch lesbar (z.B. PGP von Zimmermann und später NAI, Inc.).
- Den vielen Augen stehen tausende von Projekten gegenüber.
- Fehler im Design werden auf Code–Ebene nicht erkannt.
- Aber immerhin ist es möglich (und legal), nach Fehlern zu suchen .