Business Case:

Open Source Security

Robert A. Gehring
Technische Universität Berlin
Informatik & Gesellschaft

EUROFORUM-Konferenz "IT-Sicherheit 2003"
Hamburg, 12. November 2003

Agenda

- Prolog: ‘Marketecture vs. Architecture’ (L. Hohmann)
- Praxisbeispiele
  - Webserver: Apache vs. Microsoft IIS
  - Betriebssysteme: Open Source vs. Closed Source
- Open Source Risiken
- Fazit (OSS-Qualitäten)

‘Marketecture vs. Tarchitecture’

«Software systems can be divided architecturally along two broad dimensions. The first is the *marketecture*, or the ‘‘marketing architecture.’’ The second is the *tarchitecture*, or the ‘‘technical architecture.’’»

Luke Hohmann: Beyond Software Architecture, 2003, S. 51

- **Eine gute ‘marketecture’ ergibt nicht unbedingt eine sichere ‘tarchitecture’, und umgekehrt...**

Z.B. Microsofts Strategie...

- **Desktop-Monopol halten und schrittweise neue Märkte erobern (Server, Mobile, Content), durch:**
  - **proprietäre Technologien**
  - **‘leveraging’**
  - **‘bundling’**
  - **‘tying’**
  - **‘forced upgrading’**
  - **‘dumping’**

➜ **‘marketecture’ dominiert ‘tarchitecture’**

und die Folgen...


1) Webserver
Apache vs. MS IIS

SANS Top 20 List (2003)

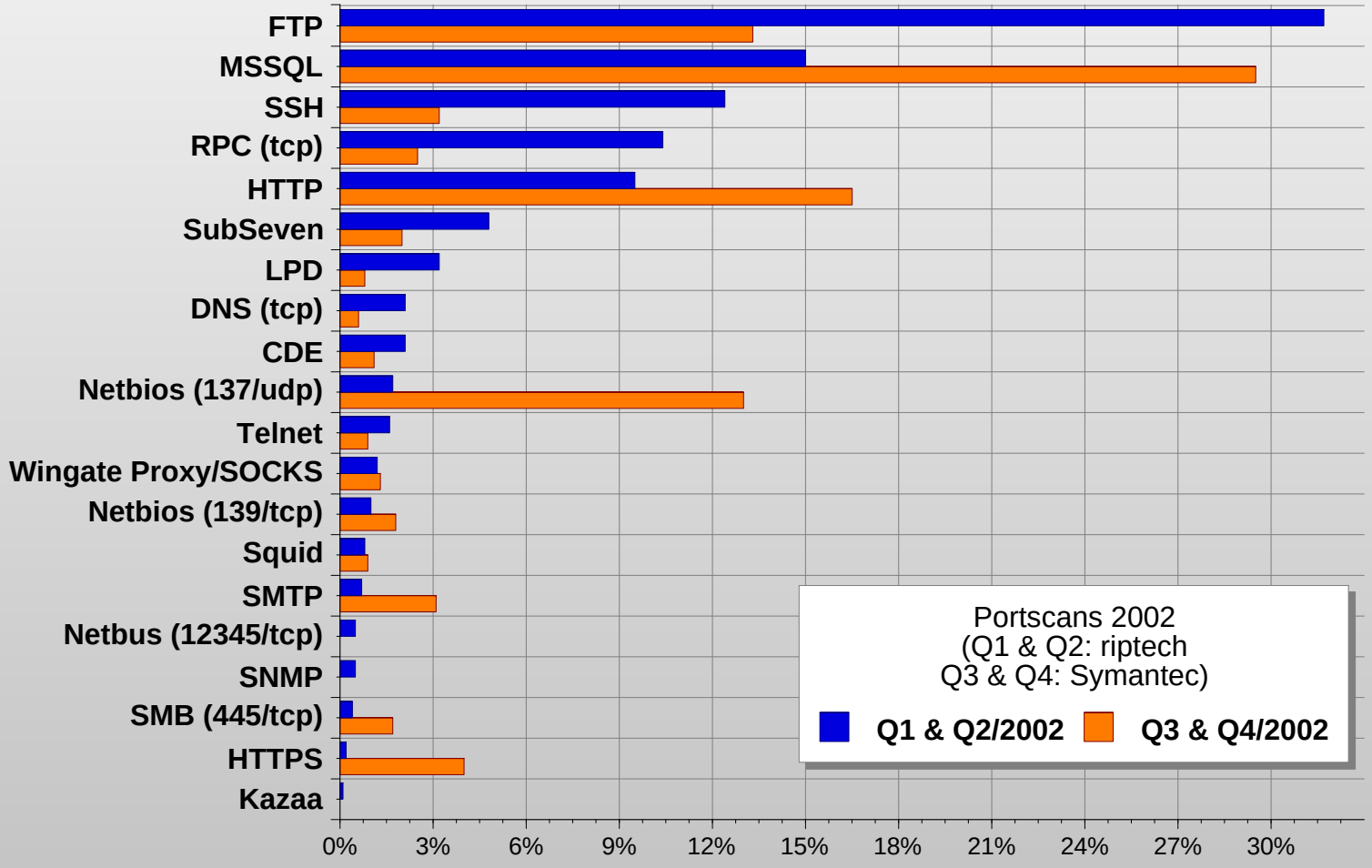
MS Windows

- **Internet Information Server (IIS)**
- Microsoft Data Access Components (MDAC)
- Microsoft SQL Server
- Internet Explorer
- Windows Scripting Host (WSH)
- **...**

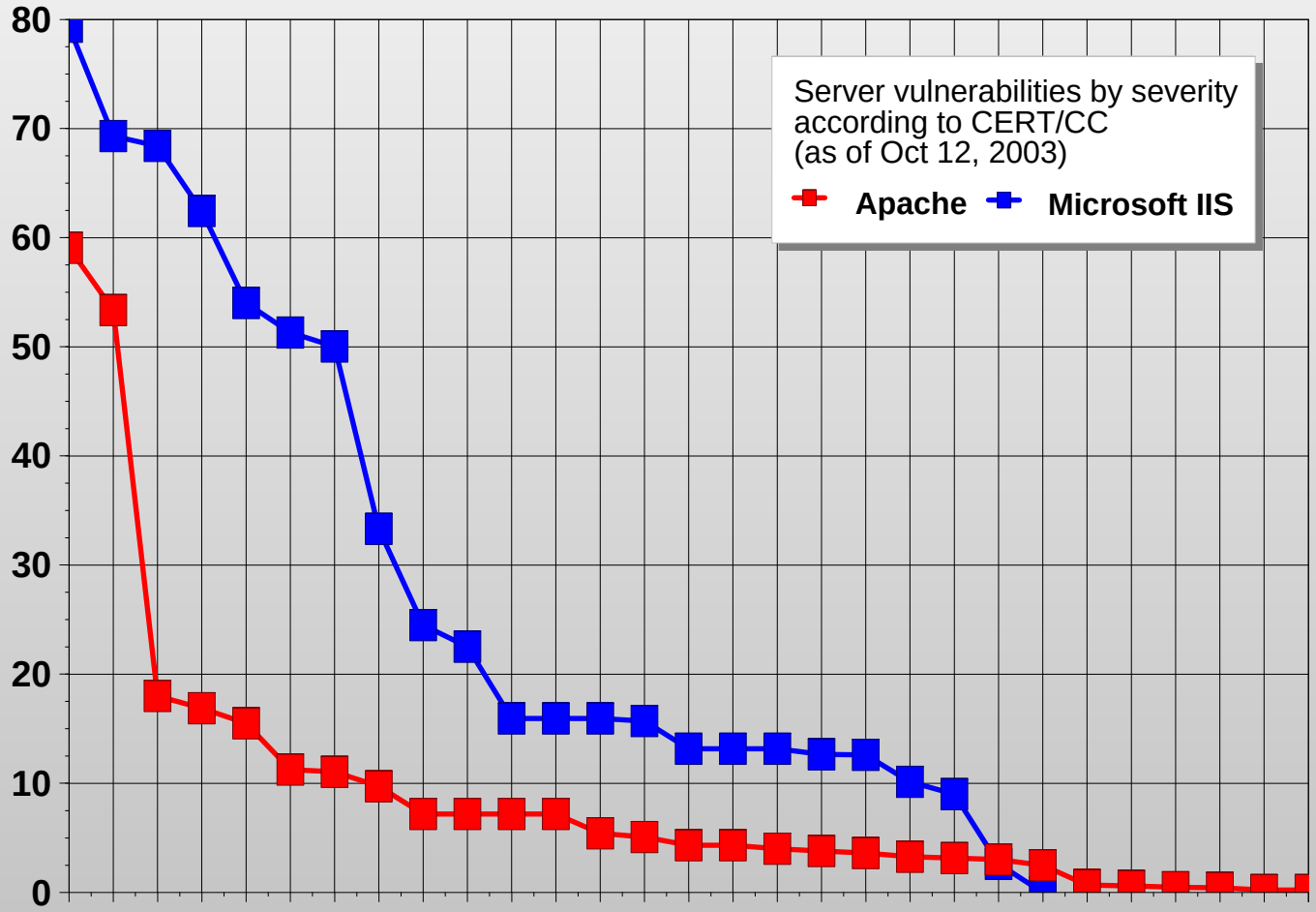
Unix

- BIND/DNS
- Remote Procedure Call (RPC)
- **Apache Web Server**
- General UNIX Authentication
- Sendmail
- ...

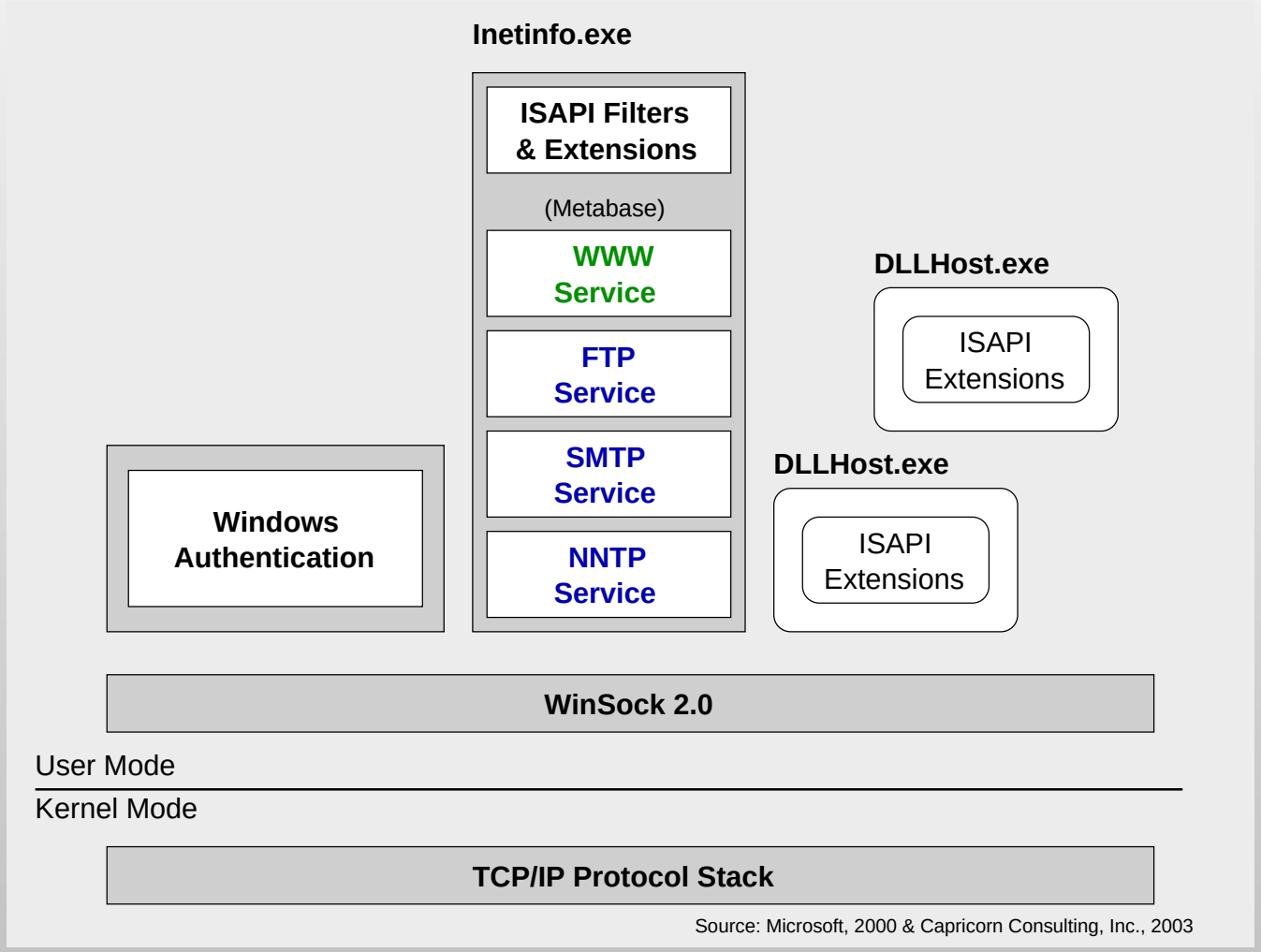
Top 20 Portscans (2002)



Webserver-‘vulnerabilities’

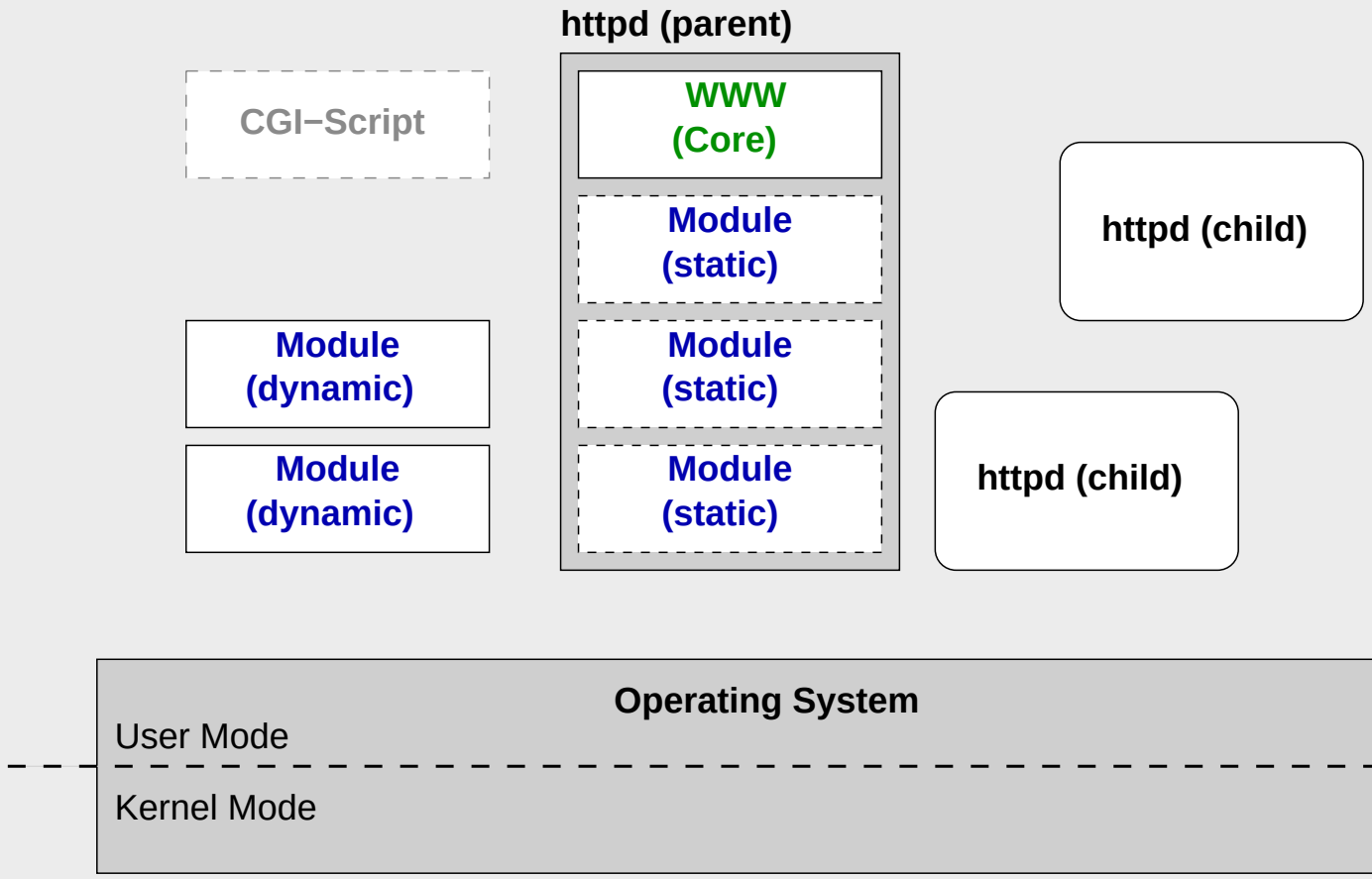


IIS 5.0 Architektur



Source: Microsoft, 2000 & Capricorn Consulting, Inc., 2003

Apache 1.3 Architektur



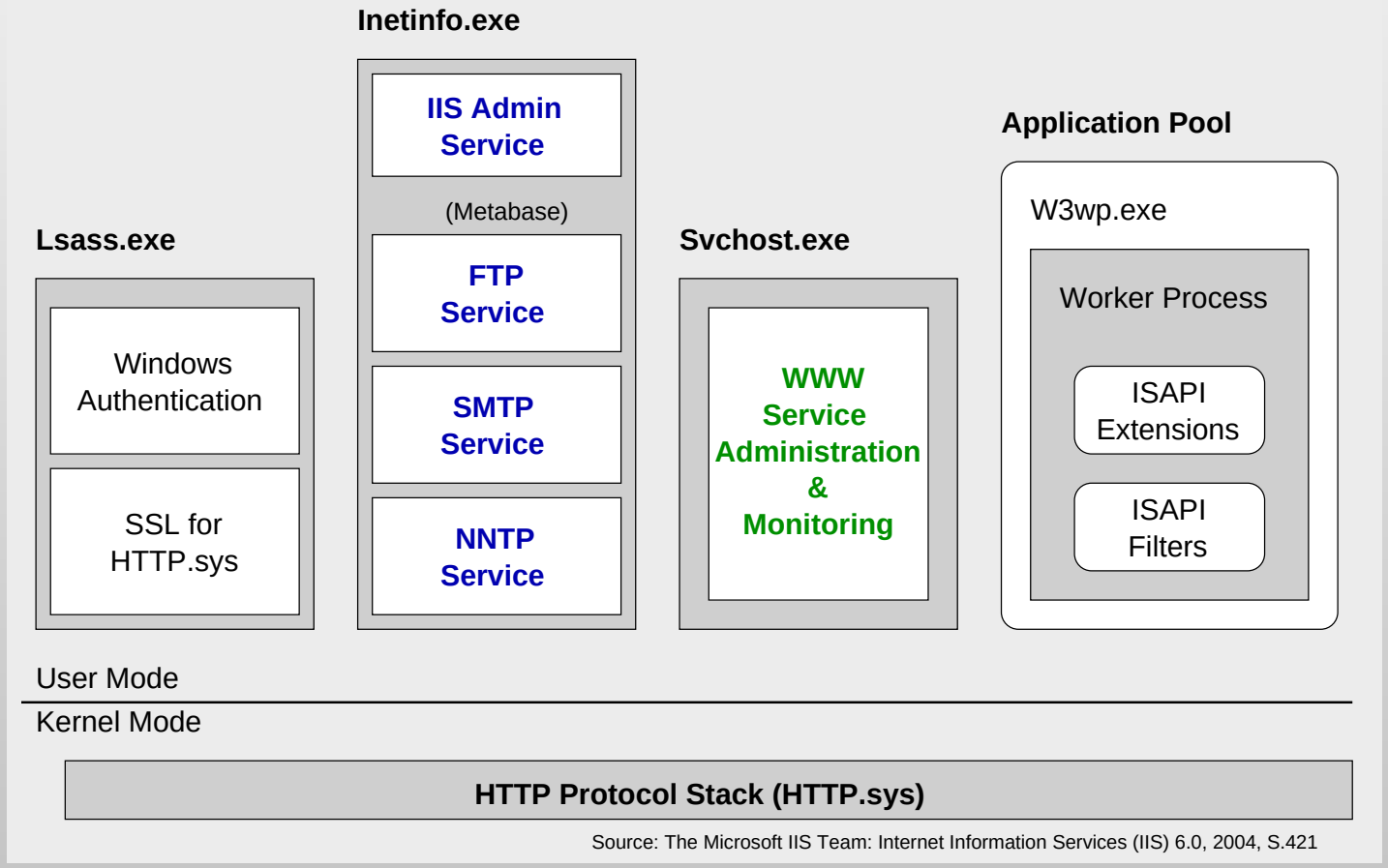
Webserver-Charakteristika (I)

	IIS 5.0	Apache 1.3
unterstütztes OS	Windows 2k	diverse Unices, NT
Server	HTTP, FTP, SMTP, NNTP...	HTTP
Aktivierung	per default bei OS−Installation	abhängig vonOS/ Distribution
Aufbau	monolitisch, ‘all in one’	modular
Privilegien	hohe	geringe bis hohe
Prozeß−separation	niedrig	mittel bis hoch
Reduktion von Komplexität	Deaktivierung Deinstallation	Deintegration Deaktivierung Deinstallation

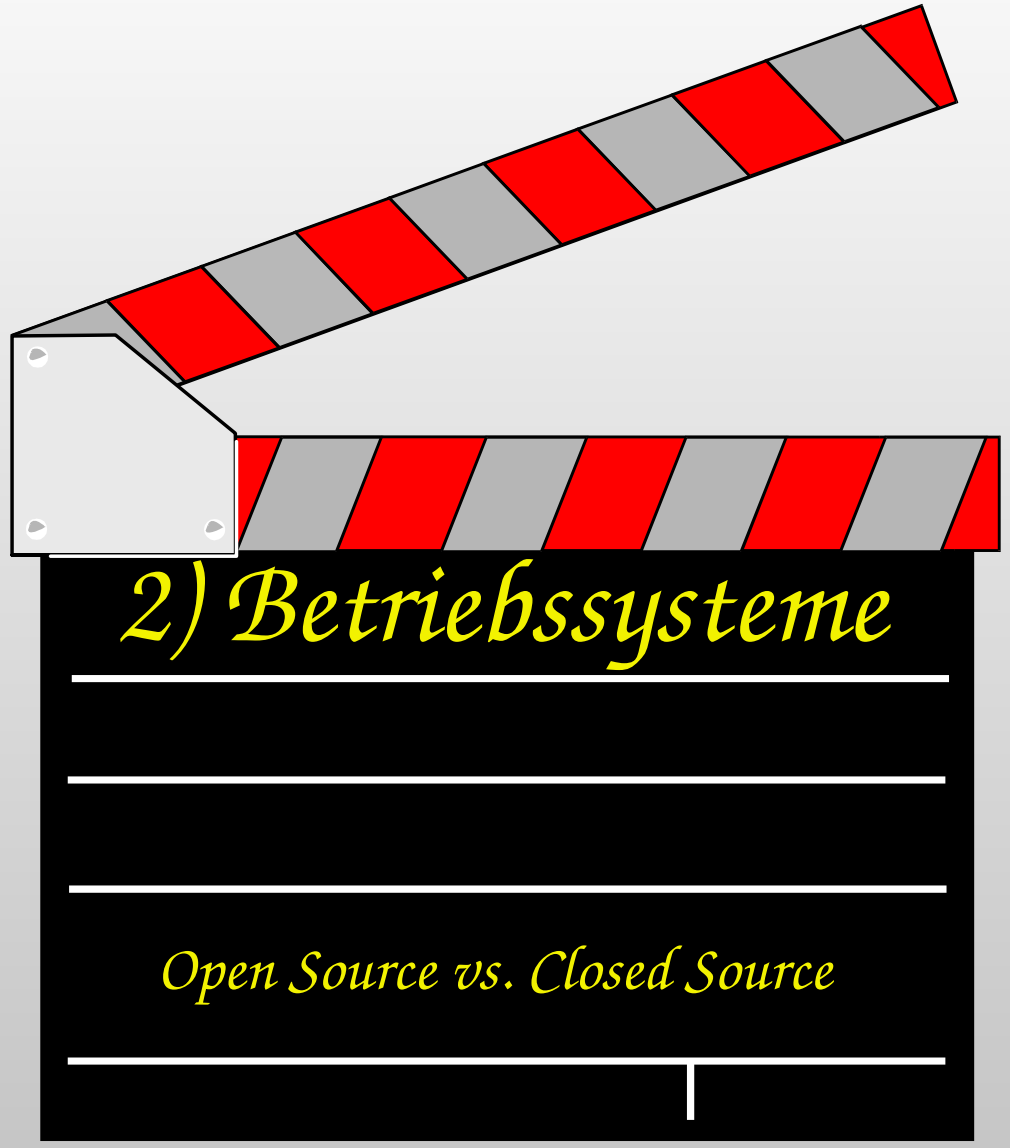
Webserver-Charakteristika (II)

	IIS 5.0	Apache 1.3
unterstütztes OS	Windows 2k	diverse Unices, NT
Server	HTTP, FTP, SMTP, NNTP...	HTTP
Aktivierung	per default bei OS−Installation	abhängig vonOS/ Distribution
Aufbau	monolitisch, ‘all in one’	modular
Privilegien	hohe	geringe bis hohe
Prozeß−separation	niedrig	mittel bis hoch
Reduktion von Komplexität	Deaktivierung Deinstallation	Deintegration Deaktivierung Deinstallation

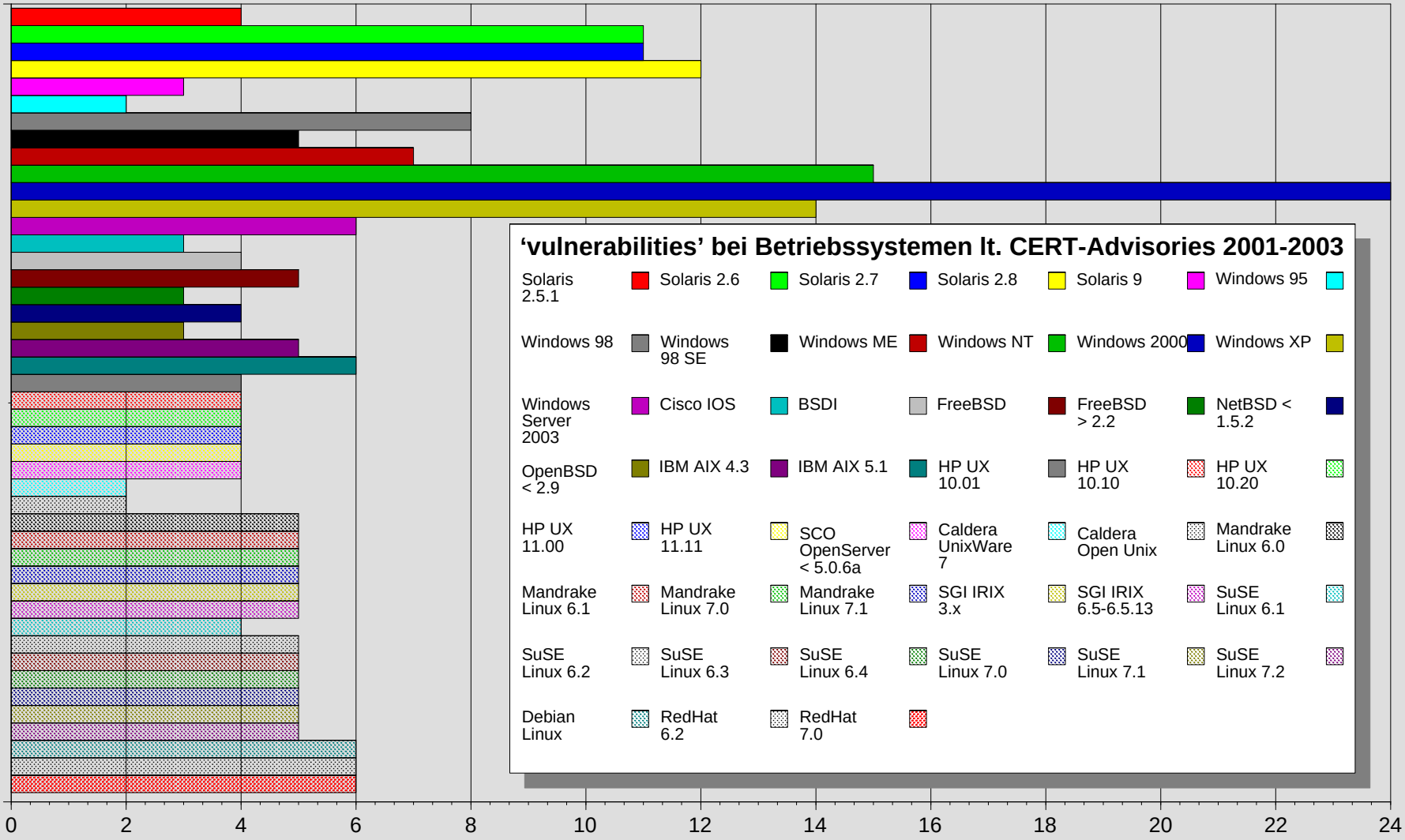
Aber: Microsoft lernt dazu ...



Source: The Microsoft IIS Team: Internet Information Services (IIS) 6.0, 2004, S.421


2) Betriebssysteme
Open Source vs. Closed Source

(Un-)Sicherheit von Betriebssystemen (*cum grano salis)*



Detailansicht *(nach C. Payne 2002)*

	Debian	Solaris	OpenBSD
Features			
Confidentiality	5.90	6.08	7.50
Integrity	5.88	6.17	7.38
Availability	7.00	5.75	6.00
Audit	6.90	5.67	7.25
Number	15	11	18
Average	6.42	5.92	7.03
Vulnerabilities			
Confidentiality	6.75	8.13	4.50
Integrity	7.70	7.40	4.25
Availability	8.10	7.00	8.00
Audit	8.33	8.42	0.00
Number	12	21	5
Average	7.72	7.74	4.19
Unscaled Score	-1.30	-1.80	2.80
Scaling Factor	1.25	0.52	3.60
Final Score	**-1.0**	**-3.5**	**10.2**

Fehlerdynamik (nach Challet/Le Du 2003)

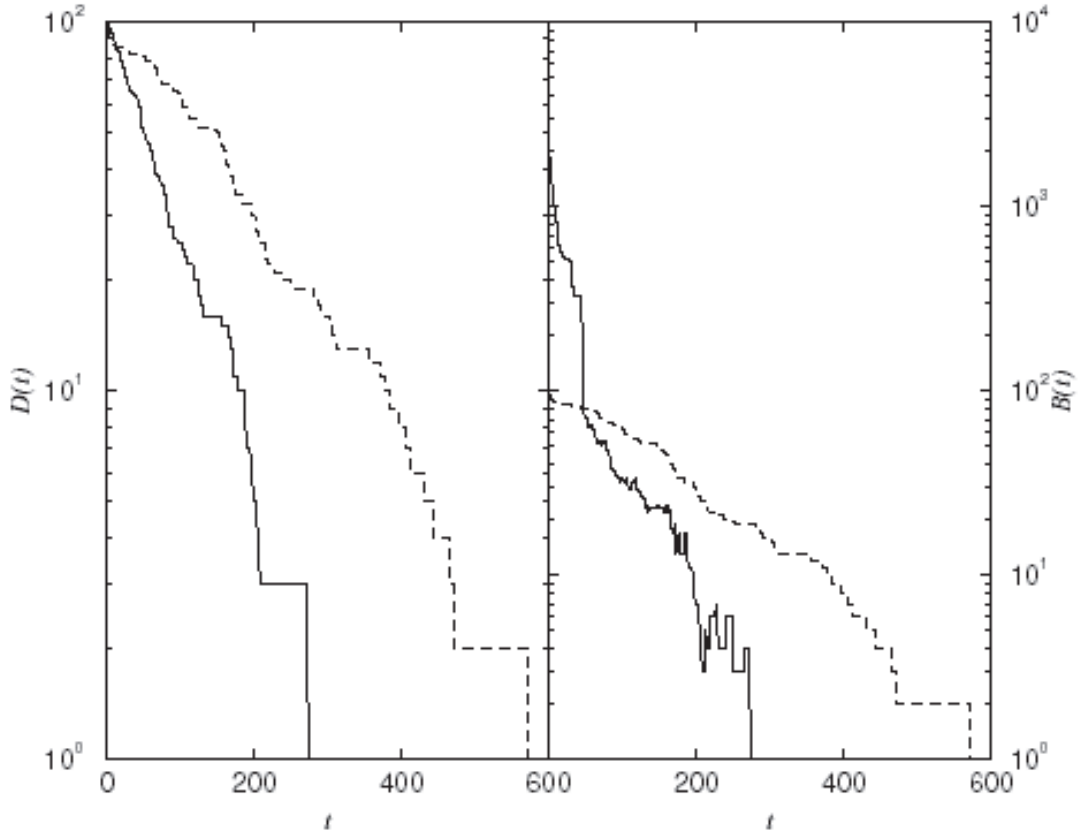


Fig. 1. Number of defective parts (left panel) and number of bugs (right panel) in an open source project (continuous lines), closed source projects with no bug report resubmission allowed between releases (dashed lines) $L = 100$, $M = 20$, $N_u = 100$, $N_p = 10$, $\delta = 1$, $\phi = 0.9$, $\beta = 0.1$, $\omega = 0.9$, $\nu = 0.9$. $T = 1$ for open source and $T = 50$ for closed source.

Eine gute Nachricht ...

«OpenBSD Release Protected Against Buffer Overflow Attacks (11 April 2003)

The most recent release of OpenBSD should eliminate buffer overflows, according to the group's project leader. The group took three approaches to hardening the software. First, the location of the stack in memory is randomized. Second, the team added a tag to the memory structure that will detect address modifications. Finally, they managed to divide the main memory into two sections: writeable and executable; the pieces of data and programs, called "pages", would be stored in one or the other section, ensuring that no page is writeable and executable at the same time.»
http://www.sans.org/newsletters/newsbites/vol5_15.php

... und ihre Bedeutung

**«[...] Many kudos are in order here.
If what the OpenBSD people are doing really works, they will put considerable pressure on other vendors and developers to do the same. Buffer overflow problems continue to plague operating systems and applications. Eliminating this category of vulnerabilities would be a major victory for the information security arena.»**

http://www.sans.org/newsletters/newsbites/vol5_15.php

(Nicht nur) Open Source-Risiken

- **Prevention:**
  - Ist die Qualitätskontrolle effektiv?
  - Werden Trojanische Pferde ‘draußen’ gehalten?
  - Ist die ‘Tarchitecture’ auf Sicherheit ausgerichtet?
  - Wurde die angemessene Lösung ausgewählt?
  - Sind die Systemverwalter kompetent?
- **Detection:**
  - Findet Peer Review *tatsächlich* statt? Kontinuierlich?
  - Kommen Sicherheitstools zum Einsatz?
  - Findet ein kontinuierliches Auditing statt?
- **Reaction:**
  - Werden Systeme gepatcht?
  - Wird die Architektur angepaßt?

Das sollte *von Fall zu Fall* betrachtet werden!

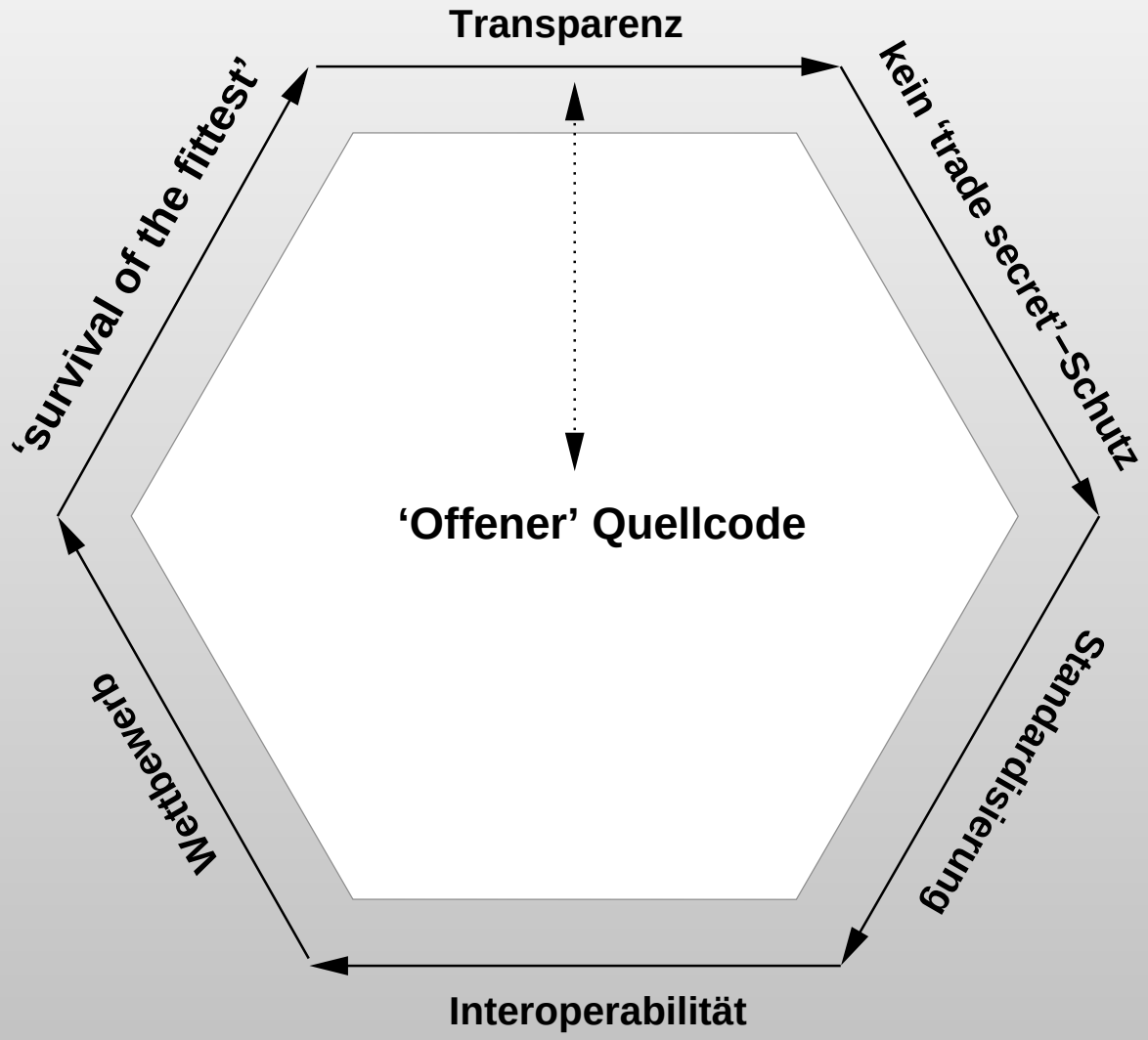
OSS vs. CS - *conclusio* (I)

- **funktionierender Wettbewerb (statt Monopol) ➜ Sicherheit als Wettbewerbsargument**
- **Entkopplung der Entwicklung von 'time-to-market'-Druck**
- **Quellcode verfügbar ➜**
  - **Transparenz: Qualitätsevaluation, Risikoevaluation mgl.**
  - **individuelle & zweckangepaßte Lösungen**
  - **Vielfalt: 'natürliches Immunsystem' entsteht**
  - **Unabhängigkeit von Herstellerstrategien ('marketecture vs. tarchitecture' -- L. Hohmann)**
  - **vereinfachte Reaktion auf unerwartete Risiken**

OSS vs. CS - *conclusio* (II)

- OSS-Entwicklungsmodell erzwingt
  - Modularisierung ➜ Reduktion von Komplexität
  - Standardisierung ➜ ‘best practices’ gefördert
- Vielfalt ➜ erfolgreiche Angriffe skalieren schlechter (‘Immunsystem’)
- ‘KISS’-Philosophie ➜ grenzt Schaden bei erfolgreichem Angriff ein

Der OSS-‘virtuous cycle’



Referenzen

Luke Hohmann: Beyond Software Architecture, Addison-Wesley, 2003.

Christian Payne: On the security of open source software, Info Systems Journal, 2002, S. 61-78.

Damien Challet & Yann Le Du: Closed source versus open source in a model of software bug dynamics, 2003, preprint: http://arxiv.org/pdf/cond-mat/0306511.

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