Open Source und IT-Sicherheit

Dirk Wetter

Head / Senior Consultant
Dr. Wetter IT-Consulting, Hamburg

info@drwetter.org

Dr. Wetter
IT-Consulting

Überblick

I. Was ist Open Source?
   - Ökonomische+Technische Gründe

II. Warum sicherer?
   - Aufklärung Märchen „security through obscurity“
   - Statistiken

III. Pen-Testing-Software

I. Was ist Open Source?

Über was rede ich:

- **Betriebssysteme**: Microsoft vs. Linux
- **Applikationen**: IE vs. Firefox, IIS vs. Apache
  MS Office vs. OpenOffice
  Oracle,MSQL vs. PostgreSQL,MySQL
- **Embedded**: Netgear vs. Linksys
- **Appliance**: Cisco, Extreme, Checkpoint vs.
  Astaro, vantronix, Fortinet

I. Was ist Open Source?

Open Source-Lizenzen:

- **O**pen **S**ource **D**efinition der OSI:
  - read (Quelltext-Einsicht)
  - redistribute (Weitergabe Prg.+Quelltext)
  - modify (Recht zur Modifikation)

I. Was ist Open Source?

Gesamtwirtschaftlicher Vorteil:

- Kosten (Lizenzen, einige TCO-Studien)

- Offenheit = Kommunikation
  - Internetprotokolle, Dokumentenstandards/-lesbarkeit

I. Was ist Open Source?

Gesamtwirtschaftlicher Vorteil:

- Offene Programmierschnittstellen = keine Monokultur
  - *„Monokulturen sind nicht nur in der Ökologie, sondern auch in der Informationstechnik eine ungute Entwicklung“ (BMI, Zypries 2001)*
  - *„Wir bekennen uns zu Offenen Standards und zur Vielfalt in der Software-Landschaft der Behörden, weil durch den Wettbewerb um beste Lösungen Qualität und Innovationen gefördert werden“* (BMI, Vogt 2004)

II. Warum ist OSS sicherer?

- Einfache Argumente
  - Marktwirtschaft
  - Flexibilität
- Märchen I: Transparenz
- „Nacktbaden“: Statistiken
- Zeit bis Fix: Statistiken
- Krypto-Märchen
- Auditierbarkeit: backdoors / covert channels

II. Warum ist OSS sicherer?

Einfach: Marktwirtschaft

- Ökonomischer Druck:
  - kein „release now, bugfix later“

    (auch: Stabilität/Zuverlässigkeit, Performance)

II. Warum ist OSS sicherer?

Einfach: Flexibilität (Achtung „technisch“):

- Quellcode: kann es selber verbessern (Bugfix)
- Neuübersetzung → (kleiner) Sicherheitsgewinn
  - Adressen verschieden
  - Anwendung anpassen, kein Feature-Ballast
  - ggf. Compiler-Härtungsflags

    (gcc: `-DFORTIFY_SOURCE, -fpie`)

II. Warum ist OSS sicherer?

Märchen 1: Transparente Software = Unsicherheit

Hersteller: „*Von mir gibt's keine Quelltexte, nur Binärprogramme. Daher weiß nur ich, wie es funktioniert, daher ist es viel sicherer*“

2 Fehlannahmen:

1.: Quelltexte geheim, daher sicherer
2.: Nur ich weiß, wie es funktioniert

II. Warum ist OSS sicherer?

Märchen 1: Transparente Software = Unsicherheit

Hacker: „*Lustig! Natürlich kann ich Binärprogramme in den Disassembler laden, und analysieren, d.h. lesen, kommentieren, Schritt für Schritt abarbeiten lassen.*“

Hersteller: „*Mag sein, aber dies [Reverse-Code-Engineering] kostet Zeit und Know-How*“

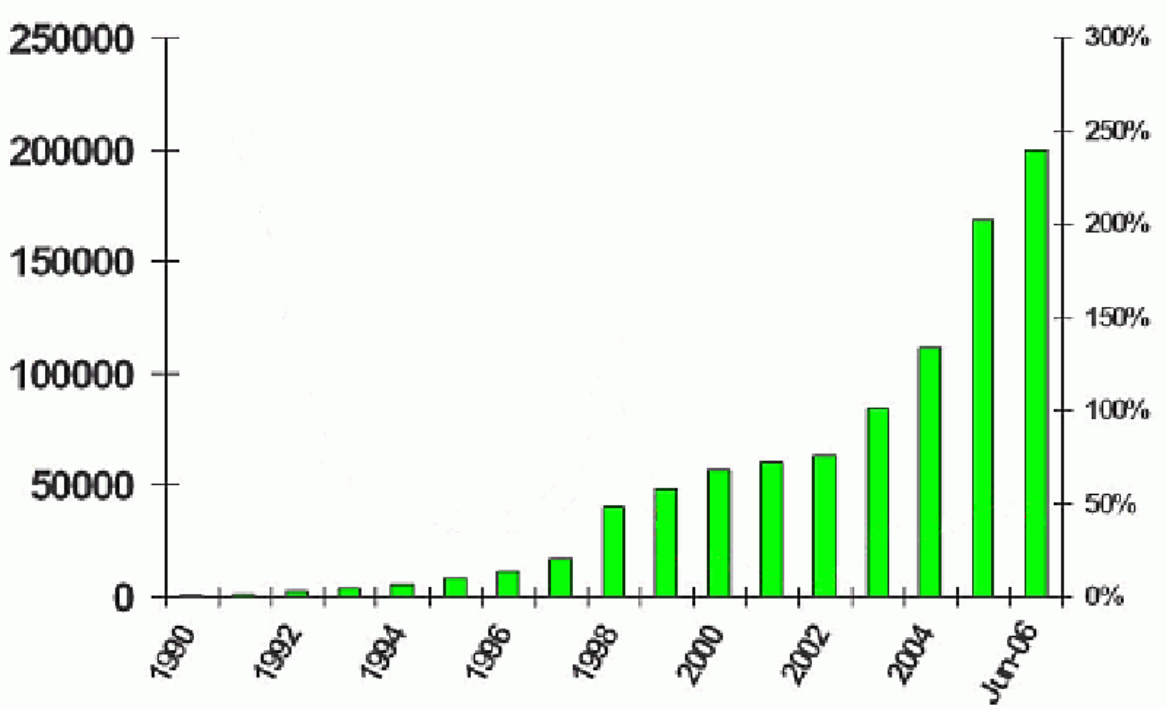
Hacker: „*Ich sehe da kaum einen Unterschied zu offenen Quelltexten, beides trifft ebenso auf OSS zu*“

Hersteller: „*Das ist aber ungleich schwieriger*“

II. Warum ist OSS sicherer?

Märchen 1: Transparente Software = Unsicherheit

Experte: „*Klar, deswegen gibt es über 100.000 Viren für MS-Software [2004 laut BBC-Studie] und ca. 40 für Linux+OSS, die meisten davon Labortest-Viren*“



McAfee Malware:

200.000 9/2006

100.000 2004

Dr. Wetter
IT-Consulting

II. Warum ist OSS sicherer?

Märchen 1: Transparente Software = Unsicherheit

- ... und was denken **Sie**, wie man ohne Kenntnis des Quelltextes
  - a) sicherheitsrelevante Fehler finden
  - b) Malware schreiben kann?

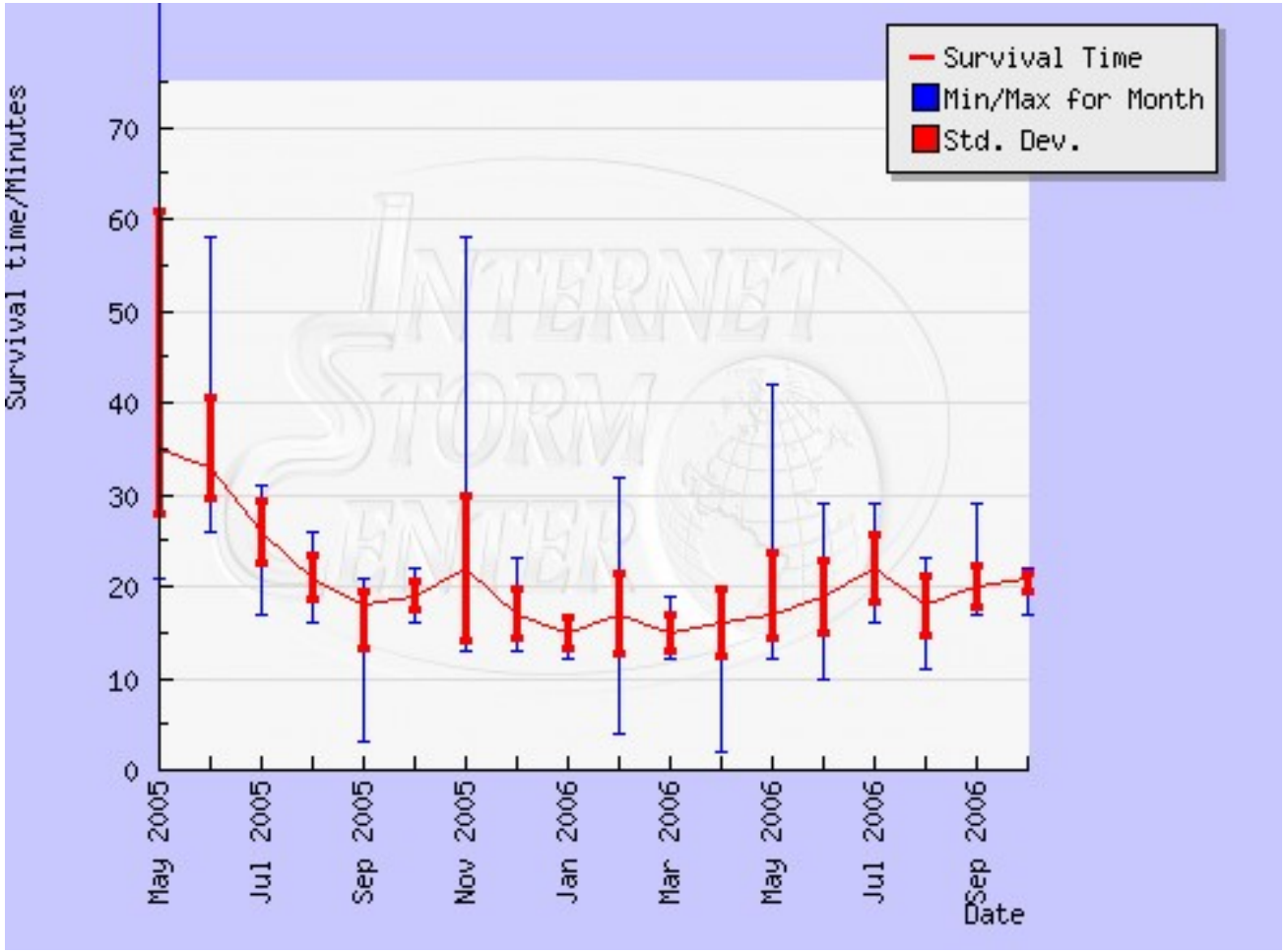
II. Warum ist OSS sicherer?

Statistiken II: Ungeschützt ins Netz („skinny-dipping“)

	Windows	Linux
Honeypot-Projekt 12/2004	15 min (pre-SP2)	3 Monate (default, schwache PW)
„Avantgarde-Studie“ (2004)	4 min (XP SP1)	keine (14 Tage)

II. Warum ist OSS sicherer?

Statistiken IIa: Überlebenstraining im Internet

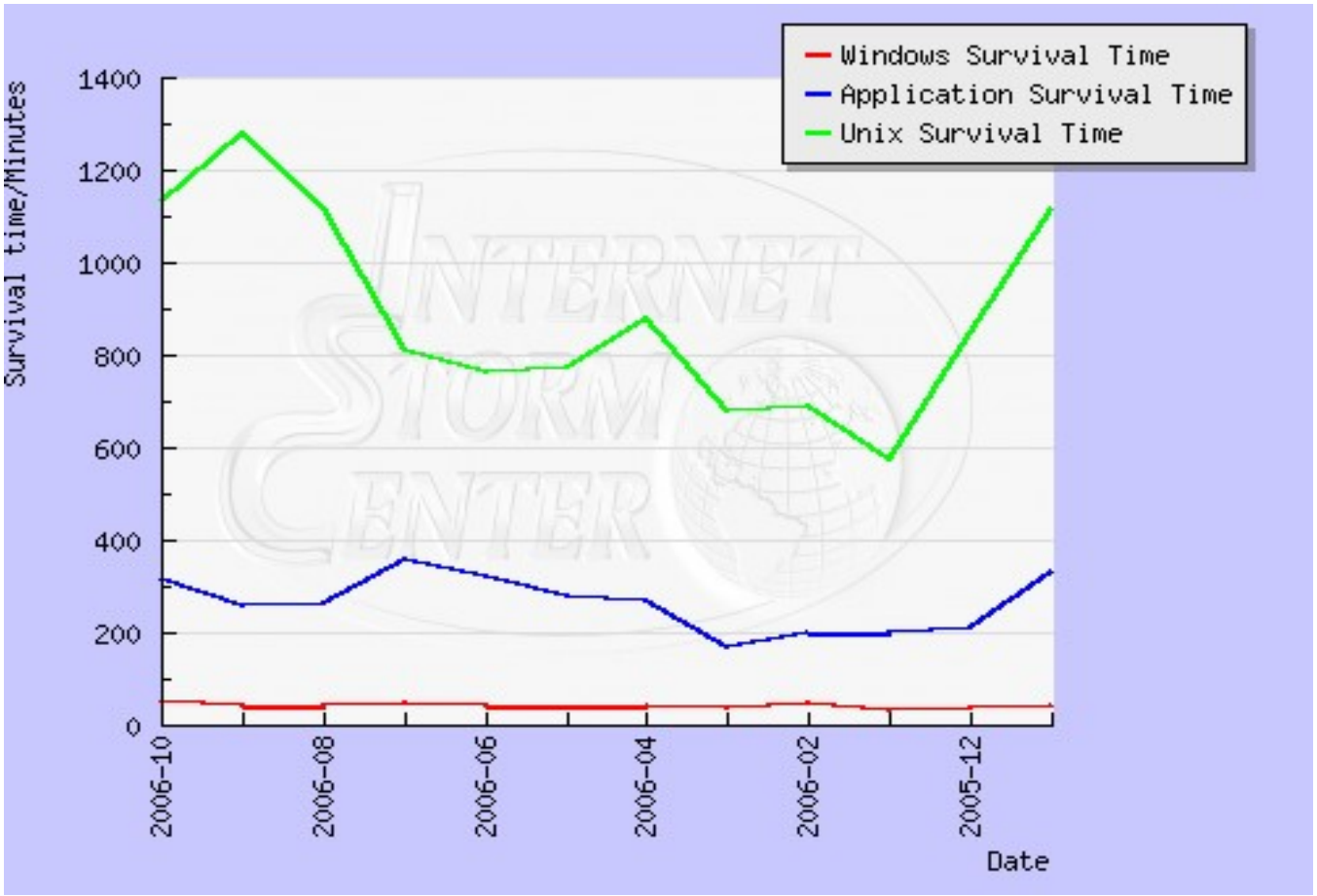


ISC of SANS (2006!)

Windows
Survival Time History

II. Warum ist OSS sicherer?

Statistiken IIb: Überlebenstraining im Internet



ISC of SANS:

Windows, Apps, Unix
Survival Time History

II. Warum ist OSS sicherer?

Statistiken II: Ungeschützt ins Netz („skinny-dipping“)

- Generell wichtig:
  - Mit Patches immer auf der Höhe
  - Gute Passwörter
- Hilft bei Windows, aber keine Rettung:
  - (mindestens Host-basierte) Firewall (XP SP2)
  - Virenschutz
- Nicht selten: Linux-Rechner ohne Firewall

  → „meine“ bis dato nicht kompromittiert

II. Warum ist OSS sicherer?

Statistiken III: Anzahl Bugs Linux vs. Microsoft

(stellenweise bei Linux ungenau: inklusive Oracle, u.a. Anwendungen)

	Windows	Linux
US-CERT 2006		
(6.9. für 2006)	127	27
2003-2006 („by metric“)	179	94
> 40	22	1 (sendmail 2003)
CVE 2006 (Einträge+Kandidaten)	187	109 (bereinigt)

II. Warum ist OSS sicherer?

Zeitspanne für Bugfix?

- Δt von Veröffentlichung bis Bugfix sollte kurz sein!
  (natürlich ebenso die Zeitspanne bis zum Einspielen):

- „Unterwelt-Hacker“ arbeiten am Exploit, egal ob OSS oder nicht
- Sobald Bug-Details bekannt sind, gibts viel mehr davon
- Häufig mit krimineller Energie, Profit winkt!

  (Kreditkarten, Bankinfo, Bot-Netz: DDoS, Spammer ...)

II. Warum ist OSS sicherer?

Zeitspanne für Bugfix?

Terminologie: „recess days“

- Statistiken?
- Ehrlich gesagt nicht einfach, besonders neue

→ Standardstudie (SecurityPortal) von 1999 (Äonen, 7 Jahre!):

	Recess Days	Total Advisories	ø per Advisory
Red Hat	348	31	11.2
Microsoft	982	61	16.1

II. Warum ist OSS sicherer?

Zeitspanne für Bugfix?

Eweek 2002 on SSL bug (platform indep.):

> „*...that made the Internet Explorer and Konqueror browsers, respectively, potential tools for stealing, among other things, credit card information*“

- KDE-Browser: Am selben Tag Bugfix
- Microsoft: Gab ein Memo heraus, und spielte das Problem herunter

II. Warum ist OSS sicherer?

Zeitspanne für Bugfix?

US-CERT 9/2004:

> „*The U.S. government's Computer Emergency Readiness Team (US-CERT) is warning Web surfers [in order] to stop using Microsoft's Internet Explorer (IE) browser.*“

- ActiveX scripting bug, (Frames, lokale Zone)
- War MS 9 Monate lang bekannt, trotzdem: Nur Workaround!
- „*When this Trojan horse runs [..] it may perform several actions, including monitoring Internet access to capture sensitive information such as logon names and passwords, [..] credit card numbers, personal identification numbers..*“

II. Warum ist OSS sicherer?

Browser-Statistiken: Anzahl Bugs

David Hammond Studie 9/2006 basierend Secunia-Meld.
(„vulnerability intelligence provider“)

	IE	Mozilla/Firefox
Historical quantity	106	46
Moderately crit.	22	16
Highly crit.	36	14
Extreme crit.	13	0
unfixed	10	4

II. Warum ist OSS sicherer?

Browser-Statistiken: Zeitspanne für Bugfix

(selbe Studie):

	IE	Mozilla/Firefox
Arithm.Mittel/Verwundbarkeit	381	93
Median/Verwundbarkeit	210	44
Arithm.Mittel/Hohe Gefährdung	115	17
Median/Hohe Gefährdung	69	23

II. Warum ist OSS sicherer?

Märchen 2: Offene Kryptoalgorithmen = Unsicherheit

(Konsolenserver-SW, USB-Vendor)

Hersteller: „*Ich habe da einen ganz sicheren Kryptoalgorithmus entwickelt in meinem Produkt, den kann niemand knacken*“

Experte: „*Wenn er denn soo sicher ist, dann leg' ihn doch offen, damit wir überprüfen können, wie sicher er wirklich ist*“

Hersteller: „*Wie bitte?? Dann sehen doch alle wie er funktioniert(/mein Passwort), dann ist mein Produkt nicht mehr sicher*“

II. Warum ist OSS sicherer?

Märchen 2: Offene Kryptoalgorithmen = Unsicherheit

→ **Unverständnis von Kryptographie!**

- **Egal welche Verschlüsselung (sym./asym.):**
  - Passwörter/Schlüssel sollten immer sicher = nicht allgemein zugänglich aufbewahrt werden (Admin-/OS-Rechte, Smartcard, Gehirn, Tresor,..)
  - kein Klartext
  - Nie im proprietären Quelltext! Solche „Algorithmen“ sind tickende Zeitbomben, die nur drauf warten, dass sie hochgehen; schlimmstenfalls durch ein simples Probieren der Zeichenketten aus dem Binary“

II. Warum ist OSS sicherer?

Märchen 2: Offene Kryptoalgorithmen = Unsicherheit

- **Auch hier: Offen = Sicherer!**

- Freie Algos (mathematisch, Implementierung):
  - z.B. AES, Twofish, 3DES, RSA
  - Freie Programme: OpenSSL/GnuTLS, OpenSSH

II. Warum ist OSS sicherer?

Märchen 2: Offene Kryptoalgorithmen = Unsicherheit

- **Moderne Kryptoalgorithmen:**

  Zahlentheorie bzw. angewandte Mathematik

- AES-Prozess der NIST (Nachfolge DES) ab Januar 1997:
  - Prozess öffentlich
  - 15 sinnvolle Einreichungen
  - Drei Konferenzen (AES 1 - AES 3)
  - November 2001: Rijndael (zwei Belgier)

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

- **Backdoor (im Code): (un)absichtliche Hintertür**

Bugtraq 2004: „*Netgear WG602 reportedly contains a default administrative account. This issue can allow a remote attacker to gain administrative access to the device.*“

Exakter: „*Any user logging in with the username "super" and the password "5777364" is in complete control of the device.*“

„Bugfix“ von Netgear: „*I can confirm that this vulnerability still exists in the latest firmware upgrade(1.7.14) for the WG602. They've simply gone and changed the username to superman and password to 21241036.*“

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

- Dgl. Cisco in WLSE/HSE, Interbase SQL-DB
  (jeweils administrative Kontrolle)
- Hartkodierte Passwörter können ohne Quellen nicht(/nur sehr schwer) geändert werden.
- Durch „Peer-Review“ kann so etwas bei OSS nicht passieren bzw. es fällt sehr schnell auf
- Fall JAP/ANON + BKA: „Crime Detection“ (2003)

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

Covert Channel = verborgene Kanäle

- Nach-Hause-telefonierende Trojaner (z.B. „key logger“)
- Nach-Hause-telefonierende legitime Anwendersoftware (iTunes, Windows-Update, ...)

→ beides nicht wirklich *covert,* einfach f. Experten nachweisbar. Wohl aber nicht für Otto-Normalanwender.

→ „covert“ vergleichbar **Steganographie**

- Frage: Kann ich mich wirklich drauf verlassen, dass in kommerzieller Software (paranoid: Hardware) keine schwer detektierbaren Kanäle drin sind?

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

- Wiederkehrende Vermutung ohne (bisherigen) Beweis:
  - Checkpoint's Firewall-Produkte haben einen covert channel, Mossad?

→ niemand will sie in arabischen Ländern haben

- Amerikanisches Cisco? Auch eher nicht.
- Araber: „Astaro? Gut!“
- Microsoft? Wer sagt mir, dass in den ganzen Paketen nach `windowsupdate.microsoft.com` nicht (versteckte) Informationen drin sind, die ich nicht im fremden Händen wissen will?

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

- Wie schaut das mit dem Flashplayer aus?

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

Oder Skype:

- „*Skype accesses the hard disk several times per minute. This can be verified by observing the HDD's activity LED, or by using a file access monitor such as FileMon*“ (Wikipedia)

- Talk @ Black Hat Europe 2006: „*Skype was made by clever people*“ „*Total blackbox. Lack of transparency.*“ „*Impossible to scan for trojan/backdoor/malware inclusion*“ „*Anti debugging technics, Code obfuscation*“

Wer versichert mir, dass Skype nicht insgeheim mein Micro anzapft, meine Daten ausliest und das irgendwo hinschickt?

II. Warum ist OSS sicherer?

Auditierbarkeit: Backdoors / Covert Channels

- Auch hier: Durch „Peer-Review“ kann so etwas bei OSS nicht passieren bzw. es fällt sehr schnell auf

III. Pen-Testing-Software

Ohne OSS läuft nichts (Achtung, Technik):

- **nmap**: schaut nach offenen Ports / Versionen von Betriebssysteme/Server-Anwendungen
- **nessus**: Verwundbarkeits-Scanner (Version 3 nicht mehr offen)
- **(ethereal)wireshark**: Netzwerk-„Sniffer“
- **hping**: Packet-Generator fürs Netz
- **dsniff**-Sammlung (auch für MITM), **ettercap**
- **John the Ripper**: Passwort-Knacker
- **RainbowCrack**: Passwort-Knacker, Tabellen

III. Pen-Testing-Software

Ohne OSS läuft nichts (Achtung, Technik):

- **Cain & Abel**: dito „Windows only“
- **sysinternals** tools (einige zumindest als Quellen): nur Windows

- fast alles unter Windows & Unix!
  - Ernst zu nehmen: unter Windows & für Windows: Retina (**eeye**)
  - [Computer-Forensik]

BMJ (Zypries): Kriminalisierung der Pentest-Tools + Anwender,
„Hackerparagraph“ 202c

Danke

Fragen?

Dr. Wetter IT-Consulting, Hamburg

info@drwetter.org

Dr. Wetter
IT-Consulting