Zur Bedeutung von Open Source in der IT-Sicherheit

Dirk Wetter
mail AT drwetter DOT org

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

Dr. Dirk Wetter

- Standbeine:
  - Computer-Forensik
  - Sicherheitsanalysen
  - Embedded Linux
- Engagements:
  - GUUG e.V.: FFG, OSDevCon, GUUG-Hamburg
  - LinuxTag e.V.
- Co-Autor Linux-Buch, ~40 Veröffentlichungen c't/iX

Überblick

I. Open Source: Definition?

II. Warum sicherer?
   - Märchenstunden
   - Statistiken

III. Trends in der IT-Sicherheit

IV. Testwerkzeuge für IT-Sicherheit

I. Definition Open Source

- Anwendungsbereiche
- OSI: Lizenzen

Über was rede ich:

- **Betriebssysteme**: Windows vs. Linux
- **Applikationen**: IE vs. Firefox,
  IIS vs. Apache,
  MSOffice vs. OpenOffice,
  Oracle,MSQL vs. PostgreSQL,MySQL
- **Embedded**: Netgear * vs. Linksys,
  Symbian, iPhone vs. Motorola* (Google-Phone?)
- **Appliance**: Cisco, Checkpoint
  vs. Astaro, vantronix

*=mit/Ausnahmen

I. Definition Open Source

Open Source-Lizenzen

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

- „viral“: GPL, LGPL
- nicht viral: BSD, MPL, Apache, CDDL, ...

II. Warum sicherer?

Marktwirtschaft

- Ökonomischer Druck:
  - kein „release now, bugfix later“
  - Firmeninterne Priorisierung: Umsatz statt Stabilität/Zuverlässigkeit, Performance
  - Firmeninternes Ressourcen-Problem
- Offene Schnittstellen/APIs:
  - Belebt technische Entwicklung
    - TCP/UDP (vs. NetBIOS)
    - OpenDocument (vs. MS Doc)
    - SMTP, HTTP

II. Warum sicherer?

Flexibilität (Achtung „technisch“):

- Quellcode: kann es selber verbessern (Bugfix)
- Neuübersetzung → (kleiner) Sicherheitsgewinn
  - Anwendung anpassen, kein Feature-Ballast
  - Adressen verschieden (keine NOP sleds)
  - ggf. Härtungsflags:
    - Compiler GCC: `-DFORTIFY_SOURCE, -fpie`
    - NX/XD-Bit (manche Kernel), Exec-Shield

II. Warum sicherer?

Transparente Software = Unsicherheit?

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

Fehlannahmen:

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

II. Warum sicherer?

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

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

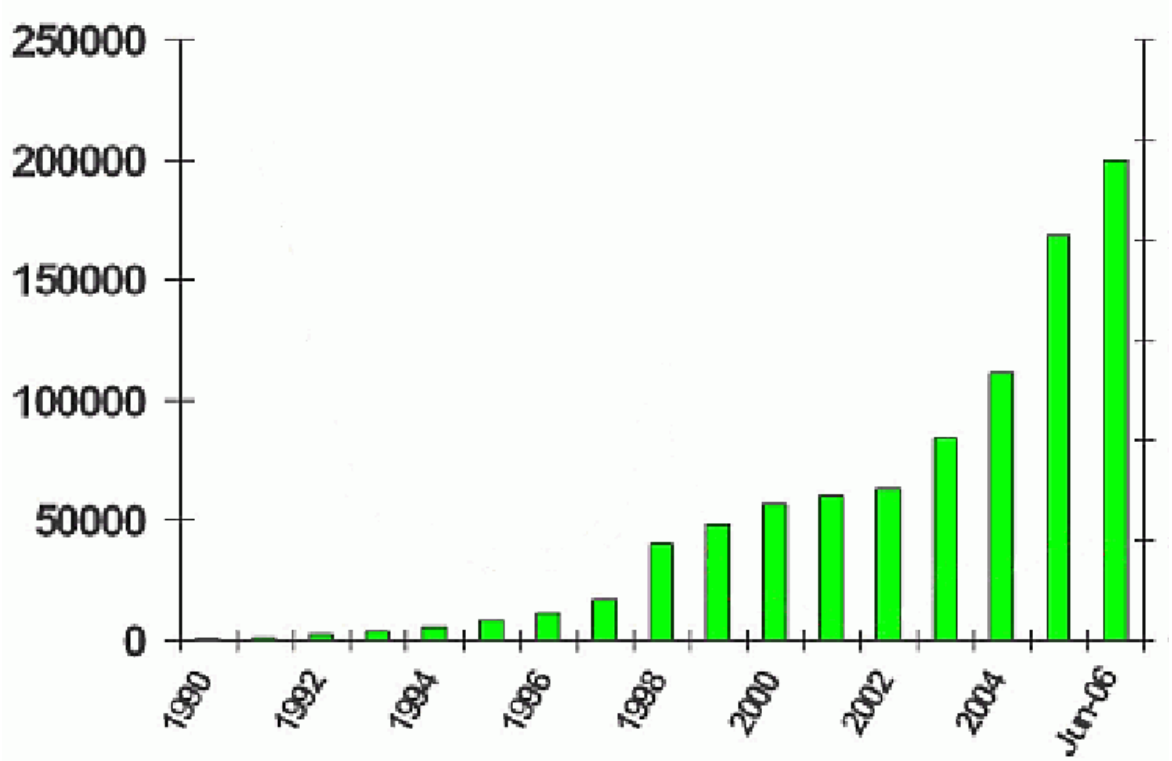
Hacker: „*Ich sehe da kaum einen Unterschied zu offenen Quelltexten, beides trifft ebenso auf OSS zu*“

Hersteller: „*Das ist aber ungleich schwieriger*“

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

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 sicherer?

- iPhone (Stand 9.10.2007)
  - Seit Erscheinen 29.6.2007 >= sechs Möglichkeiten,
    Code entfernt einzuschleusen und
    iPhone komplett zu „übernehmen“

II. Warum sicherer?

Nicht Transparent eher Unsicherheit!

- Was denken **Sie**, wie man ohne Kenntnis des Quelltextes
  a) Sicherheitslücken finden
  b) Malware schreiben

kann?

II. Warum sicherer?

Offene Kryptoalgorithmen = Unsicherheit?

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 sicherer?

- **Kerckhoffs' Prinzip! (1883)**
- **Moderne Kryptoalgorithmen:**
  angewandte Mathematik bzw. Zahlentheorie

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

II. Warum sicherer?

Offene Kryptoalgorithmen = Sicherheit!

- Kryptographie ist nicht alles!
- Sicherheit = **Kette(!),** mit potentiellen Gliedern wie
  - Designfehler
  - Fehler i.d. Implementierung
  - Vordertür dicht, „Dienstboteneingang“ offen
  - Programm sicher, Unsicherheit durch schlechte Bedienbarkeit

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Ungeschützt ins Netz („skinny-dipping“)

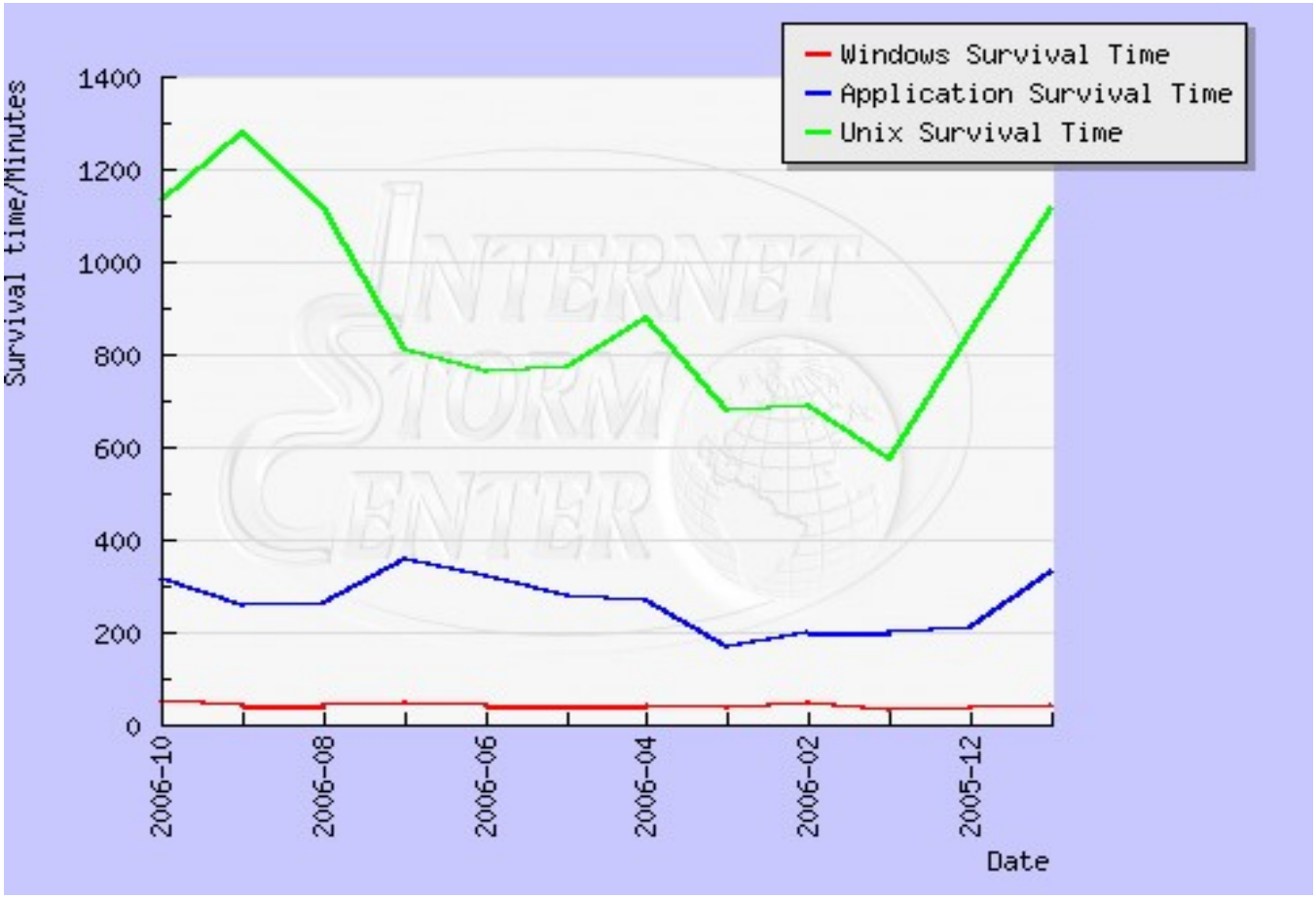
	Windows	Linux
Honeypot-Projekt 12/2004	15 Min (pre-SP2)	3 Monate (schwache PW)
Avantgarde-Studie (2004)	4 Min (XP SP1)	Keine (14 Tage)

- Nicht selten: Linux/BSD-Rechner ohne Firewall im Internet

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Überlebenstraining im Internet



ISC of SANS:

Windows, Apps, Unix
Survival Time History

Dr. Wetter IT-Consulting

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Anzahl Bugs Linux vs. Windows

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

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

II. Warum sicherer?

Zeitspanne für Bugfix?

- Terminologie: „recess days“
- Δt von Veröffentlichung bis Bugfix sollte kurz sein!
  (natürlich ebenso die Zeitspanne bis zum Einspielen)

- Cracker arbeiten am Exploit, egal ob OSS oder nicht
- Sobald Bug-Details bekannt sind, gibt's mehr Exploits
  („in the wild“)

II. Warum sicherer?

Kriminalisierung

- Gestern:
  - Viel Manuell + Massenscans
- Heute: technisch hoch automatisiert (point & click)
  - Phishing: Kreditkarten/Bankinfo, Bot-Netz: DDoS, Spammer ...
  - Mafiöse Strukturen: Profit winkt!

II. Warum sicherer?

Plattformübergreifender Bug:

Eweek 2002 on SSL bug:

> „*...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 sicherer?

US-CERT 9/2004:

> „*The U.S. government's **C**omputer **E**mergency **R**eadiness **T**eam is warning Web surfers [in order] to stop using Microsoft's Internet Explorer (IE) browser.*“

- „*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..*“
- ActiveX scripting bug, (Frames, lokale Zone)
- War MS 9 Monate lang bekannt, trotzdem: Nur Workaround!

II. Warum sicherer?

Browser-Statistiken: Anzahl Bugs

David Hammond: Studie 9/2006
(basierend auf Secunia-Meldungen)

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

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Browser-Statistiken (recess days):

(selbe Studie):

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

II. Warum sicherer?

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 for the WG602. They've simply gone and changed the username to **superman** and password to **21241036**.*“

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Backdoors / Covert Channels

- Hartkodierte Passwörter können eigentlich nicht ohne Hersteller geändert werden.
- Dgl. Cisco in : WLSE/HSE, Video Surveillance IP Gateway,
- Interbase SQL-DB
  - → jeweils administrative Kontrolle
- Durch „Peer-Review“ fällt so etwas bei OSS sehr schnell auf:
  - Fall JAP/ANON + BKA: „Crime Detection“ (2003)

II. Warum sicherer?

Backdoors / Covert Channels

- **Covert Channels = versteckte Kanäle**
  - Nach-Hause-telefonierende Trojaner (z.B. „key logger“)
  - Nach-Hause-telefonierende legitime Anwendersoftware (iTunes, Windows-Update, Google-Desktop...)

→ beides nicht wirklich *covert*: einfach f. Experten nachweisbar.

→ „covert“ vergleichbar **Steganographie [1]**

- Kann man sicher sein, dass kommerzielle Software (paranoid: Hardware) keine Covert Channels enthalten?

[1]: 28: 6,7,18,14/29,21/41/67/83

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Backdoors / Covert Channels

- Wiederkehrende Vermutung ohne (bisherigen) Beweis:
  - Checkpoints Firewall-Produkte haben einen Covert Channel, Mossad?

→ niemand will sie in arabischen Ländern haben (Astaro: „gut“!)

- Skepsis auch gegenüber amerikanischem Cisco und Windows
- Bei uns:
  - BMI fördert Einsatz OSS
  - Allerdings bestimmte kritische Bereiche z.B. Bundeswehr, Kanzlerin ;-) tun sich schwer

II. Warum sicherer?

- Einfache Argumente
- Märchen I: Transparenz
- Märchen II: Kryptographie
- Statitisken I: „Nacktbaden“
- Statitisken II: Zeit bis Fix
- Auditierbarkeit

Backdoors / Covert Channels

Skype (P2P VoIP, End2End-Encryption, acquired by ebay)

- „*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: *„blackbox“ „lack of transparency.“ „Anti debugging technics, Code obfuscation“ „Backdoor?“*

➔ Wer versichert mir, dass Skype nicht insgeheim mein Mikro anzapft, mir wichtige Daten ausliest und das irgendwo hinschickt?

II. Warum sicherer?

Backdoors / Covert Channels

- Auch hier: Durch „Peer-Review“ würden Covert Channels bei OSS sehr schnell auffallen

III. IT-Sicherheitstrends

Betriebssysteme gestern (klassisches Modell):

- Allmächtiger Administrator/root
- Benutzer etwas eingeschränkt
  - Einige Systemdateien
  - Netzwerk
  - Kernel

III. IT-Sicherheitstrends

Betriebssysteme heute:

- Alle Benutzer (inkl. Admin) weiter eingeschränkt
- Technisch:
  - MAC, RBAC:
    - RHEL: SELinux
    - SLES: AppArmor,
    - FreeBSD: TrustedBSD
    - Solaris 10U3 (TX), OpenSolaris
  - Windows Vista (MIC)

III. IT-Sicherheitstrends

Betriebssysteme heute:

- Betriebssystem sicherer
- Selbst bei
  - schlechten Admins
  - schlechte/fehlende Firmenpolicy zum zeitnahen Patchen

- Thema Sicherheit verlagert sich auf Anwendungen!

III. IT-Sicherheitstrends

Anwendungen!

- schon immer problematisch, siehe z.B.
  - Große Anwendungen: T-Hack
  - Selbst gestrickt, häufig mit wenig Ahnung:
    - personenbezogene Daten / Appliance-Hersteller / Keine Verschlüsselung/Authentifizierung, wo angebracht
- MAC/MIC hilft nicht!

III. IT-Sicherheitstrends

Herausforderungen heute:

- Web 2.0 + AJAX, Servlets usw.:
  - JavaScript / PHP u.a.: Parsing, Übergabe, Session-Handling
  - SQL-Injektionen
  - Cross-Site-Skripting (XSS): „schön“ für Foren
    - MySpace (Wurm) et. al
  - Session-Hijacking / -Riding (CSRF)

III. IT-Sicherheitstrends

Anwendungssicherheit:

- Web-Applikationsfiltern (*Web Application Firewalls*):
  Herumdoktern am Symptom? Teilweise!

- Breite Entwicklerbasis, OSS-Modell
- Sicherheit als Design, nicht später „obendrauf“
- Externe Sicherheitstests (Betriebsblindheit)

III. IT-Sicherheitstrends

Zukünftig:

- Bot-Netze: Spam, DDoS
- Verbreiterung der Angriffsfläche durch noch mehr IT
  - Mehr IT = mehr Komplexität = kein KISS [1]
  - z.B. Eintrittsvektor Handy
- Virtualisierungs-Rootkits?
- „Viel Potential“: VoIP

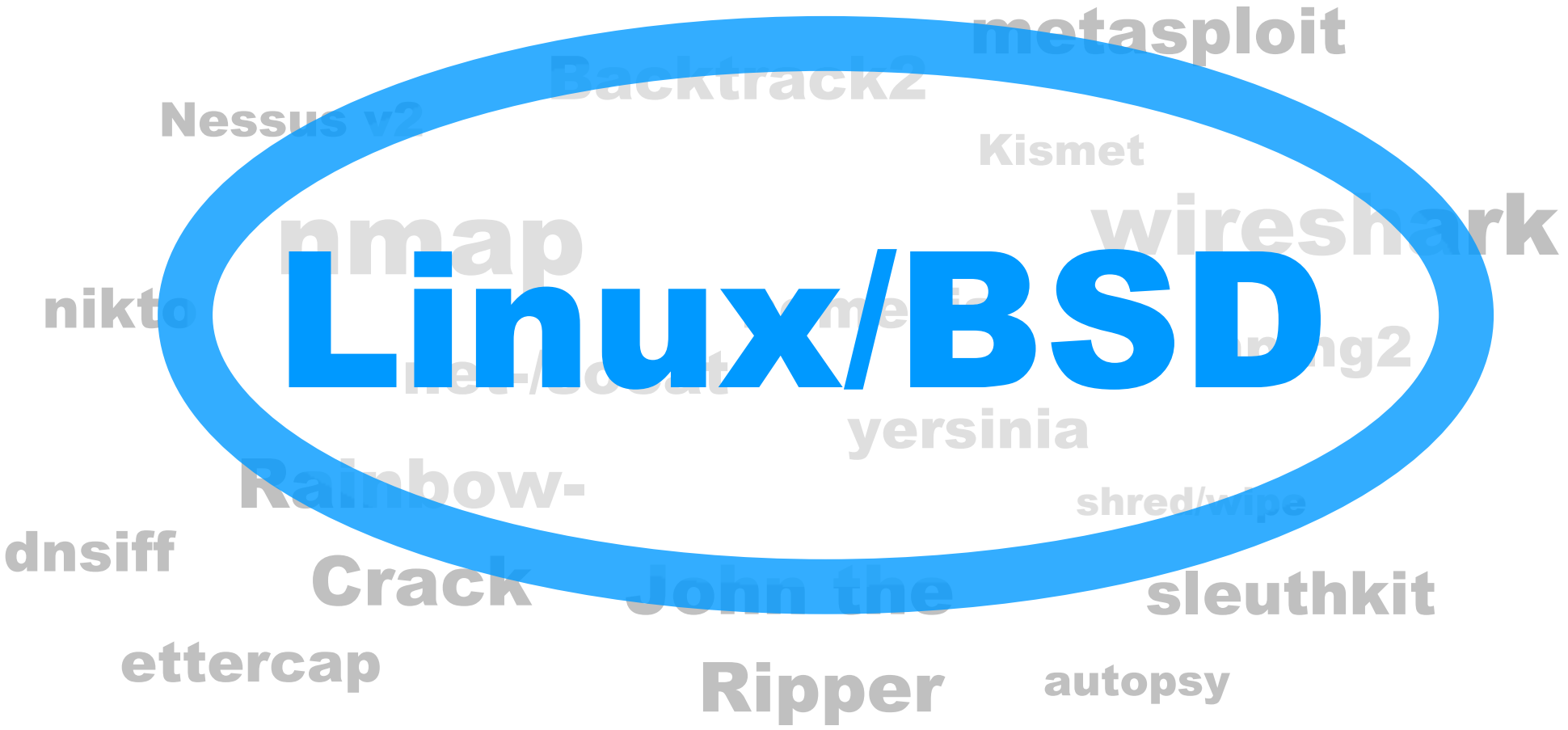
[1]: Albert Einstein

IV. Tools

Ohne OSS läuft nichts!

IV. Tools

Ohne OSS läuft nichts!



Danke für Ihre Aufmerksamkeit!

Fragen?

Dieser Vortrag wurde erstellt mit folgender OSS:
OpenSuse, Ubuntu, OpenOffice Impress, KDE u.v.a.