



# Linux vs Microsoft (Un-)Sicherheit

von

Sebastian Kinzler,

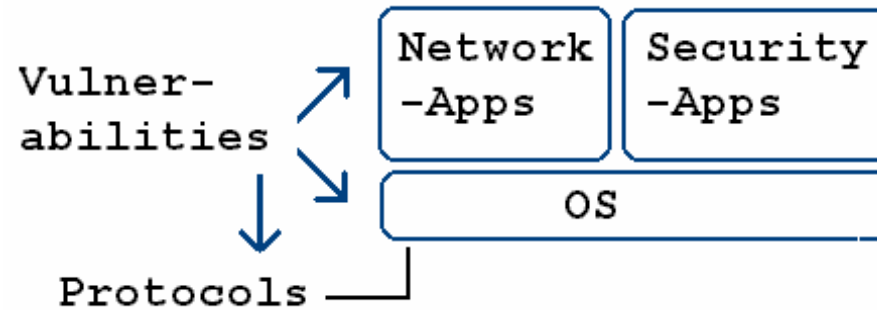
Andreas Tschersich



# Inhalt

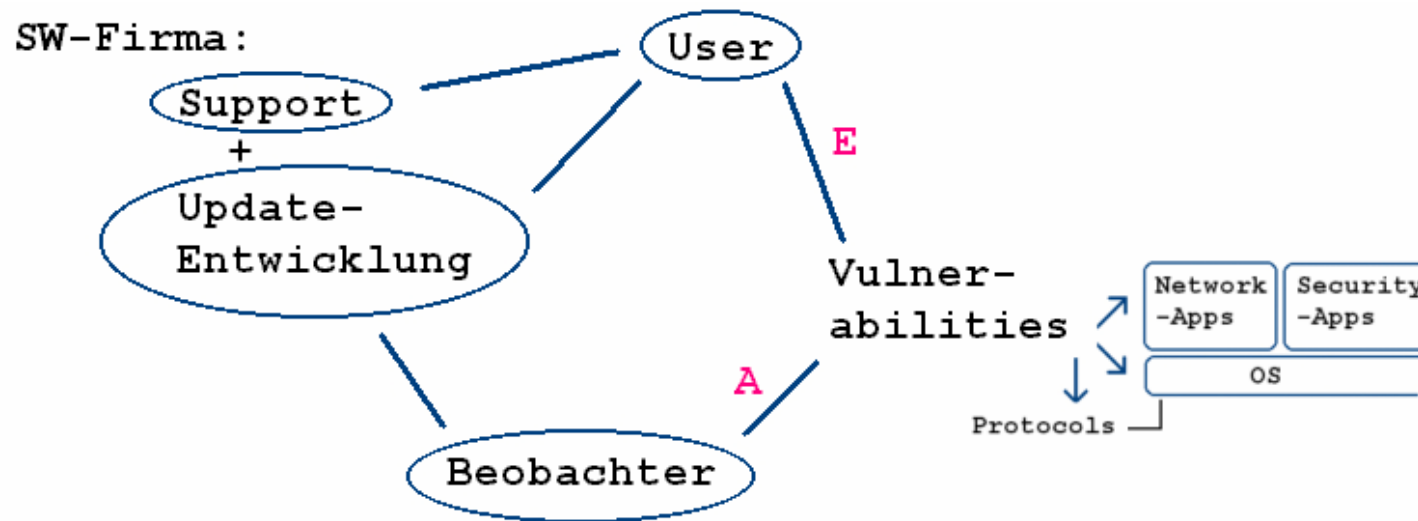
- Einleitung
- Update-Kreislauf
- Zugriffskontrolle
- Angriffe
- Sans Top 20 Unsicherheiten
- Studie von Forrester Research + Meinung von Andreas
- Closed versus Open-Source
- Studie von Nicholas Petreley
- Linux + Windows-Sicherheitskonzepte
- Meinung von Sebastian

## 1) Einleitung



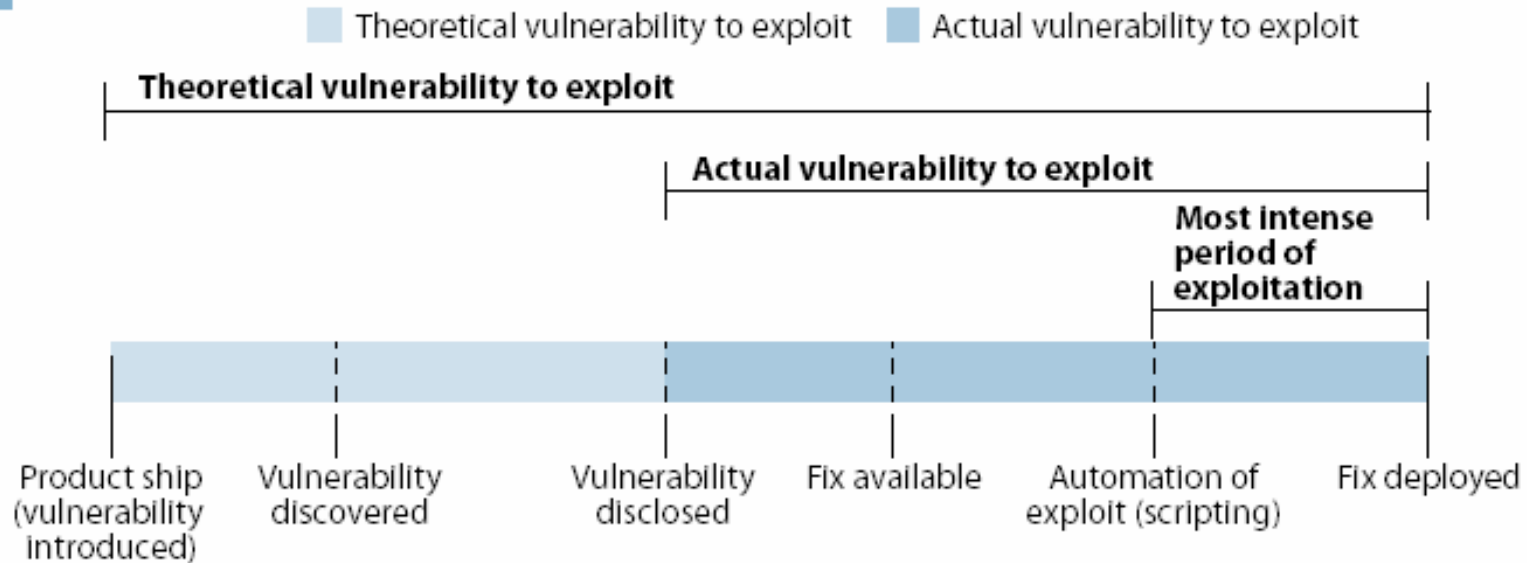
- Unsicherheiten (Sicherheitslücken) sind auf Anwendungssoftware-, Betriebssystem-, und Protokollebene vorhanden.
- Kann eine Verletzlichkeit einer Ebene durch die anderen Ebenen sowie Schutzmassnahmen nicht ausgeglichen werden, so sind erfolgreiche Angriffe möglich.
- Network-Apps interpretieren und generieren Nachrichten zu User-Zwecken.
- Security-Apps sollen bestehende Unsicherheiten aufspüren bzw. ausgleichen (entgegenwirken), z.B. Port Scanner, Firewall
- Betriebssysteme werden entweder auf Workstations (PCs) oder Serversystemen eingesetzt. Dazu gibt es spezielle Versionen. Windows XP, Suse Linux 9.1; Windows Server 2003, Red Hat Enterprise Linux AS V3

## 2) Update-Kreislauf

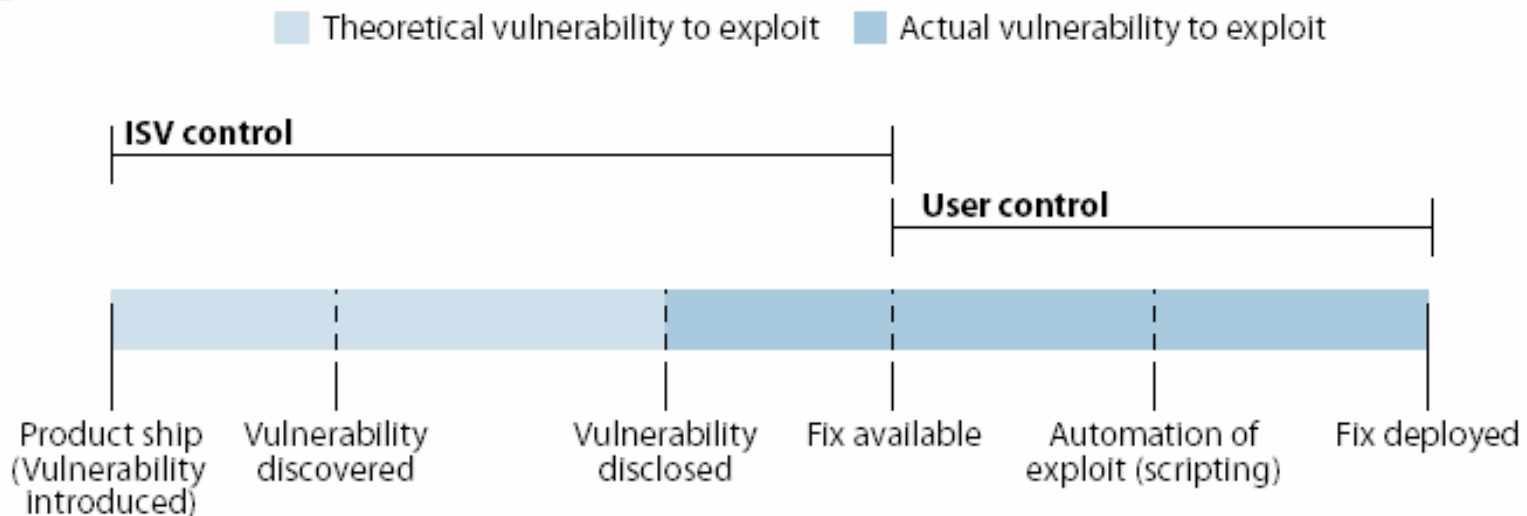


- Solange kein Update vorliegt und der Unsicherheit nicht durch andere Massnahmen entgegengewirkt werden kann, dürfte der betroffene Service eigentlich streng sicherheitstechnisch nicht mehr genutzt werden!
- Updates ändern die Konfiguration, fügen Dialoge und Funktionalität hinzu, verbessern die (Eingabeauswerte-) Logik + liefern neue Unsicherheit

## 1-1 History of a vulnerability



## 1-2 Who's responsible?




## September 2004 Security Update for JPEG Processing (GDI+)

Published: September 14, 2004 | Updated: October 12, 2004

The GDI+ security update for September 2004 addresses newly discovered issues in JPEG processing technology. This issue affects Office, and listed software. Install the update.

### Security Bulletin MS04-028

Severity	Software affected	Update number
 Critical	• Windows XP	<b>830348</b>
	• Windows XP Service Pack 1 (SP1)	<b>831931</b> <b>831932</b> <b>832332</b>
	• Windows Server 2003	<b>833987</b>
	• Windows Journal Viewer	<b>833989</b> <b>833944</b>

### SUSE Security Announcement

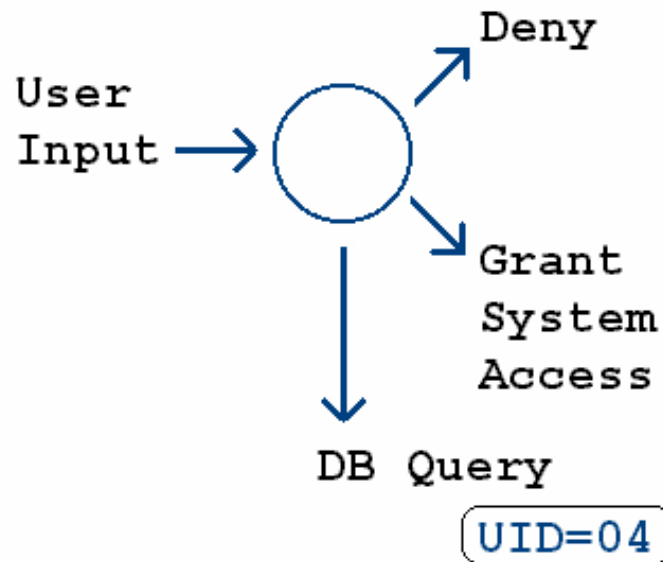
Package: xshared, XFree86-libs, xorg-x11-libs  
Announcement-ID: SUSE-SA:2004:041  
Date: Wednesday, Nov 17th 2004 15:00 MET  
Affected products: 8.1, 8.2, 9.0, 9.1, 9.2  
SUSE Linux Desktop 1.0  
SUSE Linux Enterprise Server 8, 9  
Novell Linux Desktop 1.0  
Vulnerability Type: remote system compromise  
Severity (1-10): 8  
SUSE default package: yes  
Cross References: none

Content of this advisory:  
1) security vulnerability resolved:

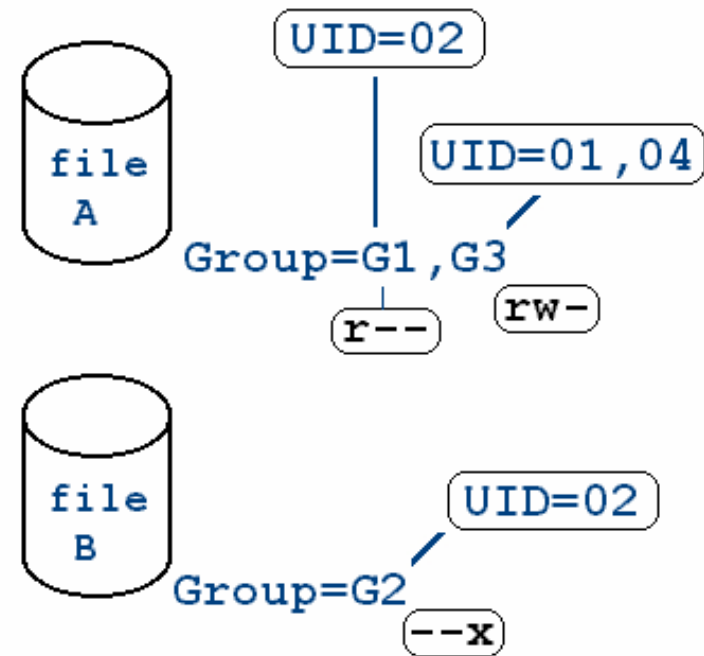
- [www.microsoft.com](http://www.microsoft.com) -> Security: Current Security Updates, Recent Incidents + Automatic Update Services
- <http://www.suse.de/de/security/> -> Security Announcements + YOU (Yast Online Update)
- Zeitdauer von der Bekanntgabe einer Unsicherheit bis zur Bereitstellung eines Updates: Microsoft 25 Tage, Linux-Redhat 57 Tage
- Wie einfach ist das System (sicher) zu konfigurieren?

### 3) Zugriffskontrolle

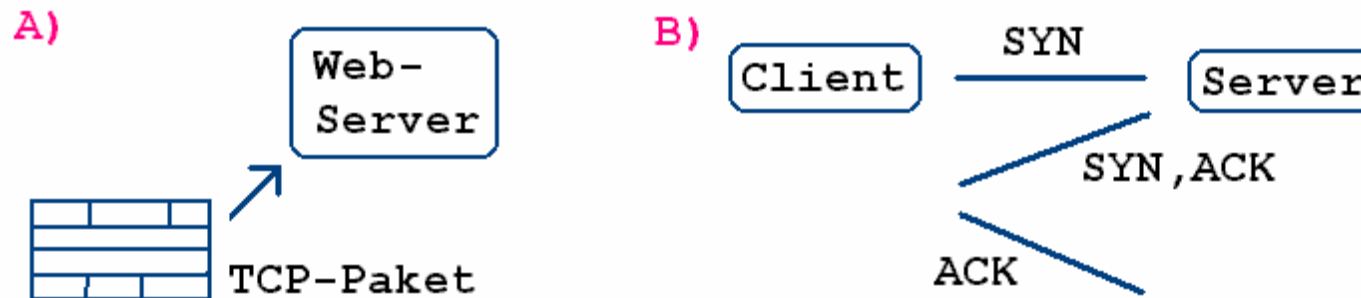
Authentication  
Procedure



Enforcing Access Privileges



## 4) Angriffe - DoS



- Ein Angreifer nutzt Pakete mit Werte-Kombinationen, die im regulären Betrieb nicht vorkommen. (Control logic performs endless looping?)
- Bei B snifft der Angreifer während des 3-Way Handshakes, liest die unverschlüsselten IP, PORT, SEQ und ACK-Werte aus und sendet unmittelbar ein RST-Paket an den Server, das noch vor dem ACK des Clients eintrifft.
- Windows NT Server: <http://www.someiismachine.com/../../../../..>  
(Null Pointer from File Handle?)

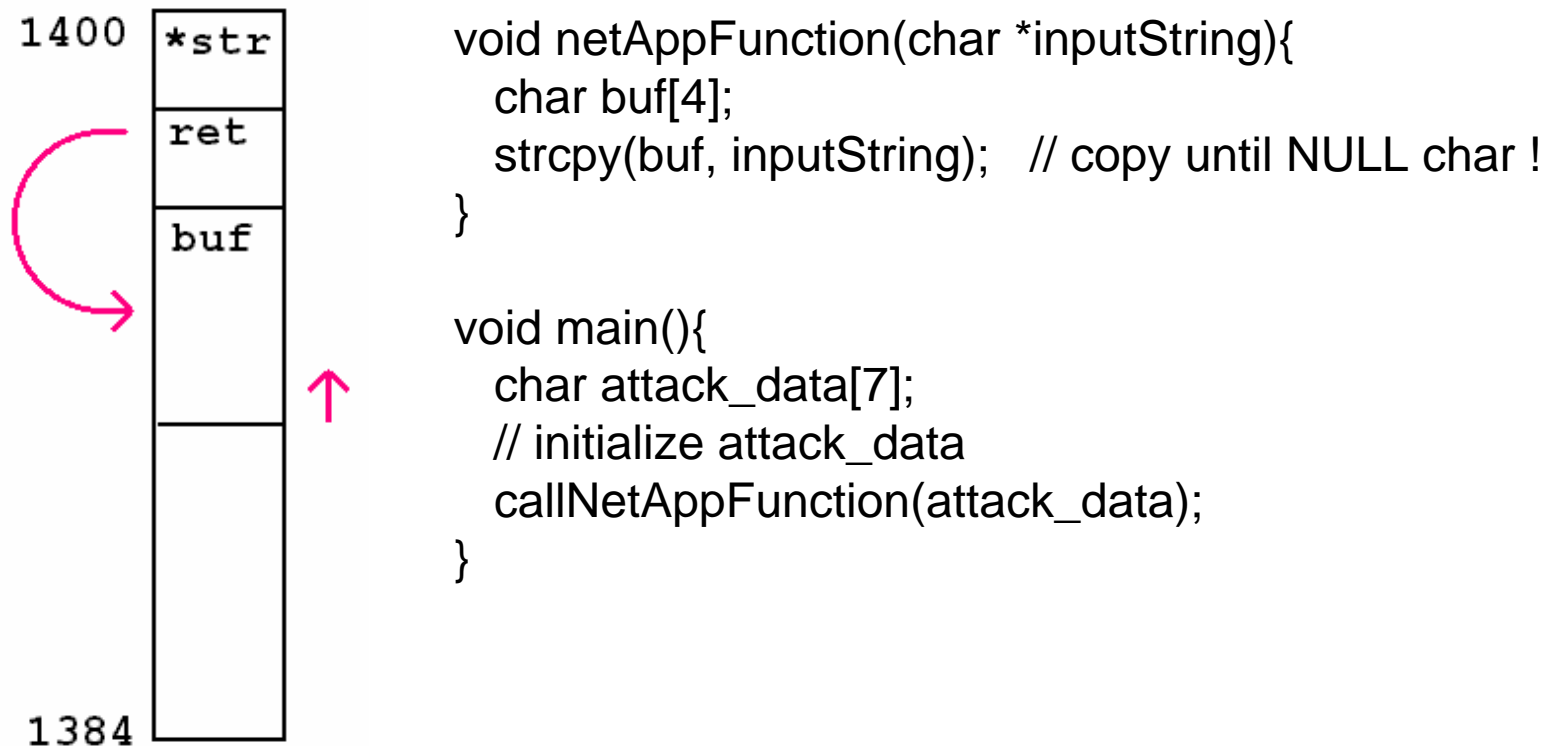




## Angriff auf Banktransaktion (Trojaner):

- Anstatt gesendete JPEG-Bilddaten zu interpretieren führt der bildverarbeitende Prozess per Buffer Overflow ein komprimiertes Executable File aus.
- Das File dekomprimiert sich in 2 Teile: Filedropper + Body
- Der Filedropper installiert eine Win32 dll unter C:\Windows\System32\ als ein BHO (Browser Helper Object) unter IE.
- Erzeugte BHO (genutzt während der Entwicklung) haben Zugang zu allen Ereignissen und Eigenschaften einer Browsing Session des IE.
  
- Bei jeder ausgehenden Kommunikation über HTTPS zu URLs mit den Strings ‚commerzbank,...‘ liest das BHO die POST/GET Daten bevor sie SSL verschlüsselt werden.
- Das BHO verschlüsselt selber die Daten, baut eine HTTP-Verbindung zum Angreifer auf und sendet die Informationen.
- Zusätzlich unterbindet das BHO jede weitere Kommunikation mit dem Bankserver.
  
- Wie viele Sicherheitslücken werden hier ausgenutzt?
- [http://isc.sans.org/presentations/banking\\_malware.pdf](http://isc.sans.org/presentations/banking_malware.pdf)
- Datagram Filtering

## Code Execution on Stack (Buffer Overflow):



- Neben Scriptausführungen eine sehr häufig ausgenutzte Unsicherheit! (Prozessrechte beachten)
- Windows XP mit SP2, Server 2003 mit SP1 und Linux unterstützen in Kombination mit den neusten CPUs (AMD, Intel) Execution Disable (XD)



## 5) Sans Top 20

- Die überwiegende Anzahl erfolgreicher Angriffe nutzt Schwachstellen ein oder mehrerer folgender Services:

- **Windows:**

- Web Servers & Services
- Workstation Service
- Remote Access Services
- SQL Server
- Authentication
- Web Browsers
- File Sharing Apps
- LSAS
- Mail Client
- Instant Messaging

- **Unix:**


- BIND System
- Web Server
- Authentication
- Version Control Systems
- Mail Transport Service
- SNMP
- Open Secure Sockets Layer
- NFS, NIS
- Databases
- Kernel

- -> Beide Betriebssysteme sind unsicher.
- [www.sans.org](http://www.sans.org) -> Top 20 List



## Sicherheits-Konzept:

- Analyse: Welcher User benötigt welchen Dienst? Welche Nachrichten müssen über welche Grenzen hinweg ausgetauscht werden?
- Entwerfen einer passenden Netzwerk-Topologie (Zonen-Aufteilung durch Firewalls + Spezialisierte Server: Mail, Web, DNS)
- Die benötigten Dienste identifizieren und nur die entsprechenden SW-Komponenten installieren.
- Dabei sichere Implementierungen und Protokolle verwenden (Firefox – IE, Apache Web Server – IIS)
- Gruppen verwenden, um Datenoperationen von Prozessen einzuschränken. (Web-Server Programm und Seiten von unterschiedlichen Usern)
- Aktivierte Dienste sicher konfigurieren (Scriptsprachen deinstallieren/deaktivieren/einschränken: ActiveX, Javascript)
- Prinzip: Alles sperren und erst bei Bedarf gezielt freigeben
- Nur vertrauenswürdige email Attachements + Downloads nutzen - Input immer auf Viren scannen
- Nur die benötigten Kommunikations-Wege per Firewall erlauben.

- 
- Tools einsetzen: Network Traffic + Access Logging, SW- und Port-Scanner verwenden (MBSA: installierte SW + Konfiguration)
  - Selber Angriffstools aus dem Internet anwenden. (PWD-Cracker)

Daraus folgt:

- Ein optimal konfiguriertes Betriebssystem mit eigentlich mehr Sicherheitslücken kann sicherer sein, als ein schlecht konfiguriertes mit ausgangsseitig weniger Sicherheitslücken.
- Die Hersteller der Betriebssysteme und Anwendungen gaben bisher der uneingeschränkten Funktionalität gegenüber der Sicherheit den Vorzug.
- Jeder aktive Service bringt aber neue Sicherheitslücken mit ein, die wieder als Ziele für Angriffe dienen können. Angreifer machen genau diese ausfindig während sie ihre Angriffe vorbereiten.

## 6) Studie von Forrester Research + Meinung von Andreas

**3-2 Percentage of high-severity flaws and of flaws fixed**

Platform	Number of total flaws	Number of high-severity flaws	% of flaws with high severity	Number of flaws fixed	% of flaws fixed
Microsoft	128	86	67%	128	100.0%
Red Hat	229	128	56%	228	99.6%
Debian	286	162	57%	275	96.2%
MandrakeSoft	199	120	60%	197	99.0%
SUSE	176	111	63%	172	97.7%

Source: Forrester Research, Inc.

- Widersprüchliche Auswertungen von Vulnerability-Report Datenbanken:  
<http://download.microsoft.com/download/9/c/7/9c793b76-9eec-4081-98ef-f1d0ebfffe9d/LinuxWindowsSecurity.pdf>

[http://www.theregister.co.uk/security/security\\_report\\_windows\\_vs\\_linux](http://www.theregister.co.uk/security/security_report_windows_vs_linux)



## Bewertung nach

- Gefahr der Ausnutzung (Experten / Basiswissen über OS und Programmierung, Mittelschwere Anpassungen an öffentlichem Basiscode, Öffentliches Angriffskit, Einfache Eingabe)
- Schadenspotenzial (Performance-Minderung; Dateien eines Users (/ Admin) lesen, verändern + löschen; Stoppen der Zielmaschine; Complete Takeover)
  
- Microsoft hat höchsten Marktanteil (Heise 2004: 16% Linux-Server in Europa; Gartner 2003: 96% aller Verkaufsgeräte mit Windows BS) -> Anreiz für Angreifer
- Keine eindeutigen Empfehlungen von CERT/DHS, BSI, NIST, NISCC

## Meine Meinung:

- Forrester: Both Windows and the four key Linux Distributions can be deployed securely.



## 7) Open Source versus Closed Source

### **Vorteile von Open Source:**

- Sicherer, da Quellcode einsehbar
  - Jeder kann ihn verbessern
  - Tausende können Fehler entdecken und beheben
- Wird immer ausführlich getestet
  - z.B. Beta-Versionen vom Linux-Kernel
  - Testszenario bei Closed Source fasst nicht erreichbar
- Programmierer nicht unter Zeitdruck
- Programmierer müssen keine Probleme ignorieren
- Sie haben nur ein Ziel
  - Erstellung eines zuverlässigen und sicheren Produktes da sie es selbst einsetzen möchten.
  - Unsichere Systeme werden nicht genutzt oder sicher gemacht





## 7) Open Source versus Closed Source

### **Nachteile von Open Source:**

- Unkoordinierte Fehlersuche und Entwicklung
- Support nicht garantiert
- Teilhabende verfügen nur zum Teil über Expertenwissen

### **Vorteile von Closed Source ( Meinung von Microsoft):**

- Organisierte Entwicklung und Fehlersuche
- Quellen stehen den Hackern nicht zur Verfügung
- Shared Source Program
  - Bei wirklichem Interesse bietet Microsoft die Möglichkeit Einblick in den Quellcode zu bekommen.
  - Grosse Unternehmen oder Universitäten können so Probleme ihrer Software bei der Integration in Microsoft-Plattformen erkennen



## 7) Studie von Nicholas Petreley

### Kritische Sicherheitslücken:

<b>Microsoft</b>	<b>Red Hat (Linux)</b>
38% ( MS-Standards ) 50% ( gleiche Standards )	10%
( CERT-Datenbank ): 39 von 40	( CERT-Datenbank ): 3 von 40 ( „Red Hat“ ) 6 von 40 ( „Linux“ )



## 8) Studie von Nicholas Petreley

### **Linux nur sicherer durch geringere Verbreitung!?:**

*Widerspruch bei der Betrachtung des Apache-Webservers:*

- Populärste Web-Server-Software im Internet
  - Laut Netcraft benutzen 68% aller Webseiten den Apache
  - Nur 21% benutzen Microsoft IIS
- Wenn Hacker wirklich das am weitesten verbreitete System angreifen würden:
  - Es gäbe mehr Würmer und Viren auf dem Apache-Webserver und damit auch auf Linux
  - mehr Angriffe gegen Apache als gegen IIS
- Wir finden genau das Gegenteil vor:
  - IIS war lange Primärziel von Würmern und anderen Angriffen
  - Sie waren auch sehr erfolgreich
  - Code Red Wurm hat mit einem Pufferüberlauf 300.000 IIS-Server infiziert
    - Verbreitung nur gestoppt da der Wurm selbst damit aufgehört hat

*Randbemerkung: Apache ist Open Source, IIS nicht*



## 8) Studie von Nicholas Petreley

### **RPC-Model:**


- Windows hängt zu stark vom RPC-Model ab
- Der Remote Procedure Call (RPC) dient der Kontrolle eines entfernten Rechners und ist damit ein potenzielles Sicherheitsrisiko
- Windows-Benutzer können RPC nicht einfach deaktivieren
  - Teilweise können RPC-Ports mit einer Firewall blockiert werden
  - Windows hängt aber so stark vom RPC-Modell ab, das dies meist nicht möglich ist
- Bei Linux ist es im Prinzip möglich alle RPC-Services zu deaktivieren und trotzdem ein funktionierenden Desktop zu haben



## 8) Studie von Nicholas Petreley

### **monolytisch oder modular :**

- Windows ist monolytisch
  - Die meisten Eigenschaften und Fähigkeiten sind in einer Einheit zusammengefasst
- Linux ist in den meisten Fällen modular
  - Die Kern-Eigenschaften und Fähigkeiten sind in eindeutige Schichten getrennt
  - Jede Schicht hat nur eingeschränkten Zugriff auf die Anderen
- Beispiel Grafikkartentreiber:
  - Bei Windows im Systemkern und bei Linux als Modul geladen
  - Ein Fehler kann bei Windows zum Komplettabsturz führen
  - Bei Linux kann ein Fehler nur den Desktop zum Absturz bringen, aber nicht das dahinter liegende System



## 9) Zusätzliche Linux-Sicherheitskonzepte

### ■ **Change-Root**

- Systemaufruf von Linux ( `chroot()` )
- Kann ausschließlich von Root verwendet werden
- Ändert das Root-Verzeichnis sämtlicher von ihm aufrufender Prozesse
- Aufgerufener Prozess weiß nicht das er im Change-Root läuft
- Das Change-Root ist also ein Unterverzeichnis innerhalb der kompletten Verzeichnisstruktur
- Hacker hackt den Prozess und steckt im Change-Root fest. Er kann damit nur Schaden im Change-Root anrichten
- Das restliche System bleibt geschützt
- Solch ein Change-Root kann auf das Mindeste an Funktionalität beschränkt werden, damit ein Hacker nur minimalen Schaden im Change-Root anrichten kann.



## 9) Zusätzliche Linux-Sicherheitskonzepte:

### ■ Paketfilter im Kernel

- Linuxsysteme besitzen seit langem einen Paketfilter
- Er ist im Kernel integriert
- Seit dem Kernel 2.4 ist der Paketfilter sogar „Stateful“ sodass bereits aufgebaute Verbindungen berücksichtigt werden können
- Mit dem Tool „iptables“ kann man dem Paketfilter komplexe Filterregeln zuweisen
- Es können damit sehr gute Firewallfunktionalitäten erreicht werden
- Damit kann ein Linux-System sehr gut geschützt werden um maximale Sicherheit zu gewährleisten



## 9) Interview mit Rich Kaplan:

### **Microsoft Vizepräsident für Sicherheitsangelegenheiten & Technologiemarketing**

- Behaviour Blocking Technologie:
  - Verdächtige Aktivitäten von Applikationen erkennen und unterbinden
    - Windows-Messenger darf keine Dateien löschen
    - Notepad soll keine Emails versenden
- Windows und Rechte von Prozessen
  - Alle Prozesse sollen nur noch mit den wirklich benötigten Rechten laufen
- Für welche Produkte und wann diese Neuerungen da sein werden ist noch offen





## 10) Meinung von Sebastian

- Linux besitzt mehr Möglichkeiten ein System zu schützen
  - Verbunden mit einer wesentlich schlechteren Bedienbarkeit
- Für ein Server-System empfiehlt sich daher ein Linux-System
- Für ein Desktop-System kann aus Gründen der besseren Bedienbarkeit auf ein Windows-System zurückgegriffen werden
- Beide Systeme müssen von vorne herein sicher konfiguriert werden und es müssen immer die aktuellsten Patches aufgespielt werden, um eine hohe Sicherheit gewährleisten zu können



■ Fragen ?