

# Abhängigkeiten in Netzwerk-Infrastrukturen

---

Abhängigkeiten in Netzwerk-Infrastrukturen  
und Auswirkungen auf produktive Systeme

Kurt Jaeger, pi@LF.net

LF.net/lf/pi

Uni Hohenheim, Stuttgart, 2. März 2004



# Übersicht

---

- . Beispiele
- . Abhängigkeiten
- . Das Backtracking Problem
- . Wer will warum ?
- . Upgrade
- . Das Für-Alle Problem
- . Das Lieferanten/Kunden Problem
- . Kennzahlen
- . Schlussfolgerungen



# Beispiele 1

---

- . HW: RAID: Firmware erkennt Plattenausfälle nicht
- . HW: Ethernet Switches, Autoconfiguration funktioniert nicht
- . Netz: Nach Update der Nameserver Software scheitert dynDNS-Update
- . Netz: Nach Umbauarbeiten in Vermittlungsstelle Leitung fehlkonfiguriert
- . Netz: In VPN: Wer darf mit wem sprechen ?
- . Netz: VPN: Endgeräte, Protokolle und Schlüsselverwaltung
- . OS: Rechner 1 steht, weil Rechner 2 runtergefahren wurde (Sparc BREAK)
- . OS: Rechner-Neustart nach zwei Jahren: Startup-Scripte scheitern



# Beispiele 2

---

- Rechte: Spamschutz, wer darf von wo Mail versenden ?
- Rechte: Wer darf welche Dateien auf dem Webserver ändern ?
- Rechte: Wer darf auf welche Datenbank zugreifen ?
- Anwendung: In welcher Reihenfolge starten ?
- Anwendung: Wer darf welchen Server sehen ?
- Anwendung: Ändern der Anbindung an ein Payment-Gateway
- Anwendung: Lizenzen und Versionen aktualisieren



# Das Backtracking Problem

---

- Update Software X ?
- Von X abhängig: Auch updaten ?
- Software A, B, C aktualisieren, von denen X abhängt
- Software Y, wie X von A, B, C abhängig ?
- software calculus
- Analog: Netzwerke und Kommunikationsbeziehungen
- Welcher Rechner/Anwendung redet mit welchen Gegenstellen/Protokollen ?
- packet calculus



# Abhängigkeiten

---

- Hardware
- Netz: IP-Adressen, Firewall, Nameserver, Domains, Protokolle, VPNs
- Betriebssystem, OS
- Rechte im Netz und auf Systemen
- Basis-Software: Webserver, Script-Sprachen, Applikationsserver, Datenbanken...
- Anwendung: Payment-Gateways, Online-Shops, CMS...
- Lieferanten



# Wer will warum einen Update ?

---

- Kunde ? Welche Abteilung beim Kunde ?
- Lieferant ? Welche Abteilung beim Lieferant ?
- Wartungsgründe ?
- Neue Funktionen ?
- Systemsicherheit ?



# Sicherheitsgründe ?

---

- . Angreifbar aus dem Netz oder Lokal ?
- . Kompromitierbar oder Denial of Service ?
- . Warten auf Anforderung oder sofort aktiv werden ?
  - Update ?
  - Filtern ?
  - Deaktivieren ?



# Das Für-Alle Problem

---

Ein Kunde, eine Anpassung, auf einem System.

- Auch für alle anderen Systeme des Kunden ?
- Auch für alle anderen Kunden ?
- Wenn nein: Wie verwalten ?
- Wenn ja: Was tun, wenn anderswo Probleme auftreten ?
- Analog bei Hardware, Netzen, OS, usw.



# Das Lieferanten/Kunden Problem

---

Beide müssen prüfen, was der andere tut.

- Zu viele Varianten, daher: Die eigene Kombination meist ungetestet
- Welche Tests sind notwendig ?
- Wie lange dauert so etwas ?



# Kennzahlen ?

---

- . Prozesszeit vrs. Arbeitszeit
- . Wieviele Personen fassen hin ?
- . Wieviele Institutionen ?
- . Wieviele Hierarchien sind involviert ?
- . Wieviel Zeit braucht der reine Durchlauf,  
z.B. für eine Rückfrage
- . Wie macht man dies vergleichbar ?
- . Wie stabil ist der Ablauf ?
- . Wie stark ist der Ablauf parallelisierbar ?
- . Wie gut ist der Ablauf testbar ?



# Schlussfolgerungen

---

- . Wie produktiv sind IT-Systeme eigentlich noch ?
- . Auch in der IT gibt es einen Schmetterlingseffekt
- . Wer haftet ?
- . Wer zahlt ?
- . Bilder sagen mehr als 1000 Worte
- . Kalkulieren ?

