IT-Sicherheit mit freier Software

Johannes Hubertz

hubertz-it-consulting GmbH

ECO e.V.

Arbeitskreis Sicherheit
Frankfurt, den 29. Mai 2006

Gliederung

- Einleitung: Vorstellung, Übersicht
- OpenOffice und Sicherheit?
- Verfügbarkeit, Vertraulichkeit, Datenintegrität
- Netzwerksicherheit

Einleitung: Johannes Hubertz

- 1954 in Köln-Lindenthal geboren
- 1973 Abitur in Köln-Mülheim
- Studium der Elektrotechnik an RWTH und FH Aachen
- ab 1980 bei großem europ. IT-Hersteller
- ab 2002 bei europ. IT-Dienstleister
- seit 1973 Bundesanstalt THW
- seit 2001 Segeln, am liebsten auf Salzwasser

Einleitung: Johannes Hubertz

- 1984 Entwicklung Sonderprodukte, Assembler, PLM
- 1988 Erstkontakt mit Unix (SCO-Xenix) und C
- 1994 Erstkontakt mit IP
- 1996 Xlink, root@www.bundestag.de, . . .
- 1997 SSLeay, ipfwadm mit shell-scripts
- 2001 Gibraltar, FreeSwan, iptables . . .
- 2001 Erste Gedanken und Konzept zu sspe

Einleitung: hubertz-it-consulting GmbH

Etwas Erfahrung war Voraussetzung

- Gründung am 8. August 2005
- Sitz: Köln
- Geschäftsinhalt: Dienstleistungen im Umfeld der IT-Sicherheit
- Logo: Johannes Hubertz Certificate Authority als ASCII-7Bitmuster
- Diese paar Bits findet sich in einigen 10000 X.509 Anwenderzertifikaten in der Seriennummer wieder
- Wir sind käuflich ;-)

Erfahrungen müssen nicht immer selbst erlitten werden

- Bellovin and Cheswick: Firewalls and Internet Security
- Fazit: Keep it simple!
- Oder mit Einstein: So einfach wie möglich, aber nicht einfacher!

IT-Sicherheit: Versuch einer Ortsbestimmung

Vier Dimensionen der Sicherheit in der Informationstechnologie

- Vertraulichkeit
- Integrität
- Verfügbarkeit
- wirtschaftlicher Aufwand

Freie Software: Versuch einer Ortsbestimmung

Freie Software mit Pflichten

- Public Domain
- Open Systems, Open Source
- Unterschiedlichste Lizenzmodelle
- Freie Software – Frei wie in Freiheit . . .
- Free Software Foundation: GNU General Public License
- Pflichten: Weitergabe mit Quellen, Urheberrecht

- Freie Software

Softwarequalität ist auch ein Sicherheitsfaktor

Ein Artikel in der Computer Zeitung Nr. 21 / 22. Mai 2006, Seite 21:
Im Opensource-Umfeld hebt das Ethos das Niveau

Hobbyfrickler sind ein Mythos

. . . Es ist übrigens nicht so, daß viele Hobbyprogrammierer an Opensource-Projekten arbeiten. Die weitaus überwiegende Zahl der hier aktiven Programmierer sind hoch qualifizierte IT-Profis. . . .

Qualitätsbewußtsein

. . . Die Qualität von quelloffener Software belegen auch unabhängige Studien. So hat jüngst Reasoning, ein Anbieter automatisierter Softwareinspektions-Services, die TCP/IP-Implementierung von Linux mit fünf kommerziellen Versionen verglichen. Das Resultat lautet, daß die Umsetzung im Linux-Kernel deutlich weniger Fehler aufweist als die der verglichenen Betriebssysteme proprietärer Herkunft.

Lars Herrmann, Solution Architekt, Redhat/fg

$<$ *JOKE* $>$Wollen wir irgendwann Lizenzen für die Gene unserer Eltern und Großeltern bezahlen, weil jemand ein Patent auf unsere Erbinformation hat?$<$ /*JOKE* $>$

Freie Software: Beispiele für Beiträge zur Verfügbarkeit

- Linux High Availability
- Apache mod_security
- OpenOffice.org

OpenOffice.org

- Kostenlose Alternative zu bekannter, proprietärer Bürosoftware
- $\longrightarrow$ISO/IEC 26300[1] wird auch übermorgen noch gültig sein!
- freies Dokumentenformat $\rightarrow$ Applikationsunabhängigkeit!
- freies Dokumentenformat $\rightarrow$ beliebige Datenkonvertierung möglich
- freie Programm-Quellen $\rightarrow$ Dokumente sind auch übermorgen noch lesbar
- freie Programm-Quellen $\rightarrow$ Sicherheit vor Hintertüren
- $\longrightarrow$ Meine Daten gehören mir!

Bezugsquelle: http://www.openoffice.org/

[1] `http://www.heise.de/newsticker/meldung/72668` vom 3.5.2006

Apache mod_security

Intention

Feingranulare Einstellung (mit regex) dessen, was der Webserver servieren darf. Exaktheit verhindert den Systemeinbruch, der Mißbrauch wird also stark erschwert.

Sorgfalt führt zum Ziel

Fehlende Angriffsfläche führt zu unterbrechungsfreiem und damit längerem Betrieb.

Linux High Availability

sind 99,999 Prozent genug?

Linux-HA ist schon einige Jahre alt. **heartbeat** ist das zentrale Element.

Targets

- Dateisysteme
- Serverfunktionalität
- Netzwerkfunktionalität

Verfügbarkeit 100 Prozent?

Manager haben **Träume**

- Wir haben eine Firewall, da ist alles sicher.
- Software installieren, fertig.
- Software installieren, einrichten, . . .
- Software installieren, einrichten, regelmässige Wartung, . . .
- Albtraum: Systemausfall!

Managers **Reality-Show**

- Verfügbarkeit ist de facto immer kleiner als 100 Prozent
- 99 Prozent bedeutet 3,6 Tage Ausfall im Jahr!
- 99,9 Prozent bedeutet 0,36 Tage = 9 Stunden
- Hochverfügbarkeit ist i.a. teuer!
- Linux-HA gibts kostenlos

Vertraulichkeit, was ist das?

Manager haben **Träume**

- Ich bin Ihr Systemadministrator, wie war doch gleich Ihr Passwort?
- Daten sind manchmal sehr wichtig.
- Verschlüsselung schützt, vorbeugen mußt Du!
- GnuPG, OpenSSL, OpenSSH, OpenVPN, cacert.org . . .

Managers **Reality-Show**

- Verschlüsselung contra Verfügbarkeit
- Wer darf an meine Daten?
- Wer darf nicht an meine Daten?
- Schlüssel sollten verschlüsselt abgelegt werden!
- Schlüsselverwaltung z.B. mit OpenLDAP als PKI
- SmartCard hilft, GnuPG kostet nichts.
- Software gibts umsonst, das KnowHow kostet . . .

Datenintegrität

Lokale Integrität

- Sind Ihre Daten in 10 Jahren noch die gleichen?
- Wie stellen Sie das sicher? Hält Ihr Programm so lange?
- Kryptographische Hashfunktionen können helfen
- Backup, Archivierung, Regelmäßige Tests . . .

Integrität beim Transport

- Datenübertragung mit Checksummen (TCP/IP)
- Offengelegte, herstellerunabhängige Standards (RFCs)
- Verschlüsselung kann zusätzliche Sicherheit herstellen

Sicherheit

Sicherheit als Balanceakt

- Verfügbarkeit und Vertraulichkeit sind offensichtlich Gegensätze
- Integrität über lange Zeiten macht Vertraulichkeit zunichte
- Eine gesunde Mixtur der drei Anforderungen ist anzustreben.
- Der wirtschaftliche Aufwand beschränkt und balanciert die Realisierung
- Ist Vertraulichkeit beim OnlineBanking auch überflüssig ???
- Muß GdPDU-konforme Email-Archivierung den Kostenrahmen sprengen?

Gliederung

- Einleitung: Vorstellung, Übersicht
- OpenOffice und Sicherheit?
- Verfügbarkeit, Vertraulichkeit, Datenintegrität
- Netzwerksicherheit

Netzwerk, wo brauche ich das?

- Mehr als ein Computer?
- Netzwerk verbindet!
- IP <=> Internet Protokol, RFC 791, September 1981
- Zu Hause, in der Firma, unterwegs, in der Raumfahrt . . .
- immer und überall mit allen Fehlern

IP-Filterung

Paketfilterung

- Linux 2.0: ipfwadm, Juli 1996
- Linux 2.2: ipchains, Januar 1999
- Linux 2.4: iptables (netfilter), Januar 2001
- Linux 2.6: iptables (netfilter), Dezember 2003

Firewalls

einfache Firewall-Lösungen mit Linux mit und ohne Grafik

- Shorewall, Ipcop, Firewall-Builder, u.v.a.m.
- TIS Firewall-Toolkit, smtpd, squid, apache, bind, . . .
- satan, nessus, sara, nmap, snort, autopsy, . . .

zentral gesteuerte Firewalls

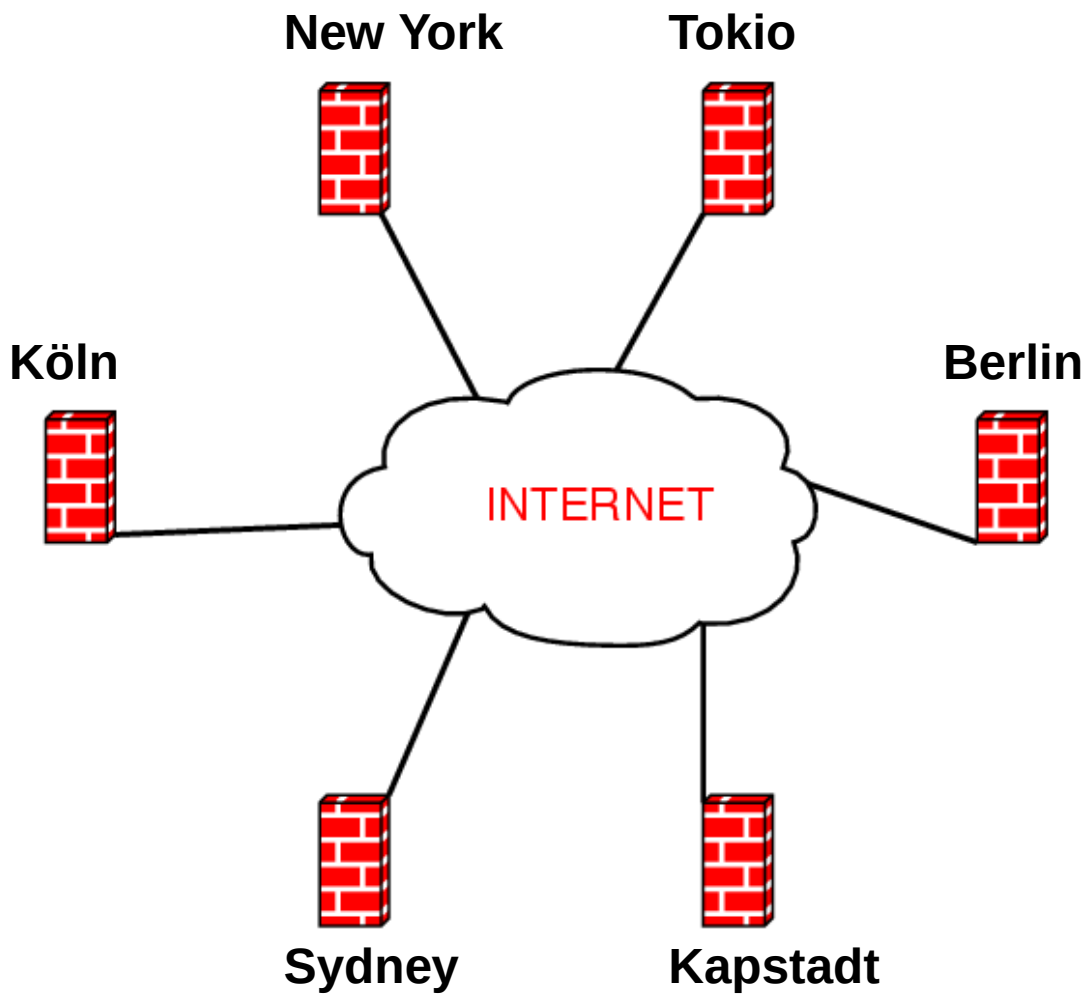
Firewall-Lösungen für komplexe Umgebungen

- iscs.sf.net, Integrated Secure Communications System
- NetSPoC.berlios.de, a Network Security Compiler
- sspe.sf.net, simple security policy editor

Übersicht: Lizenz

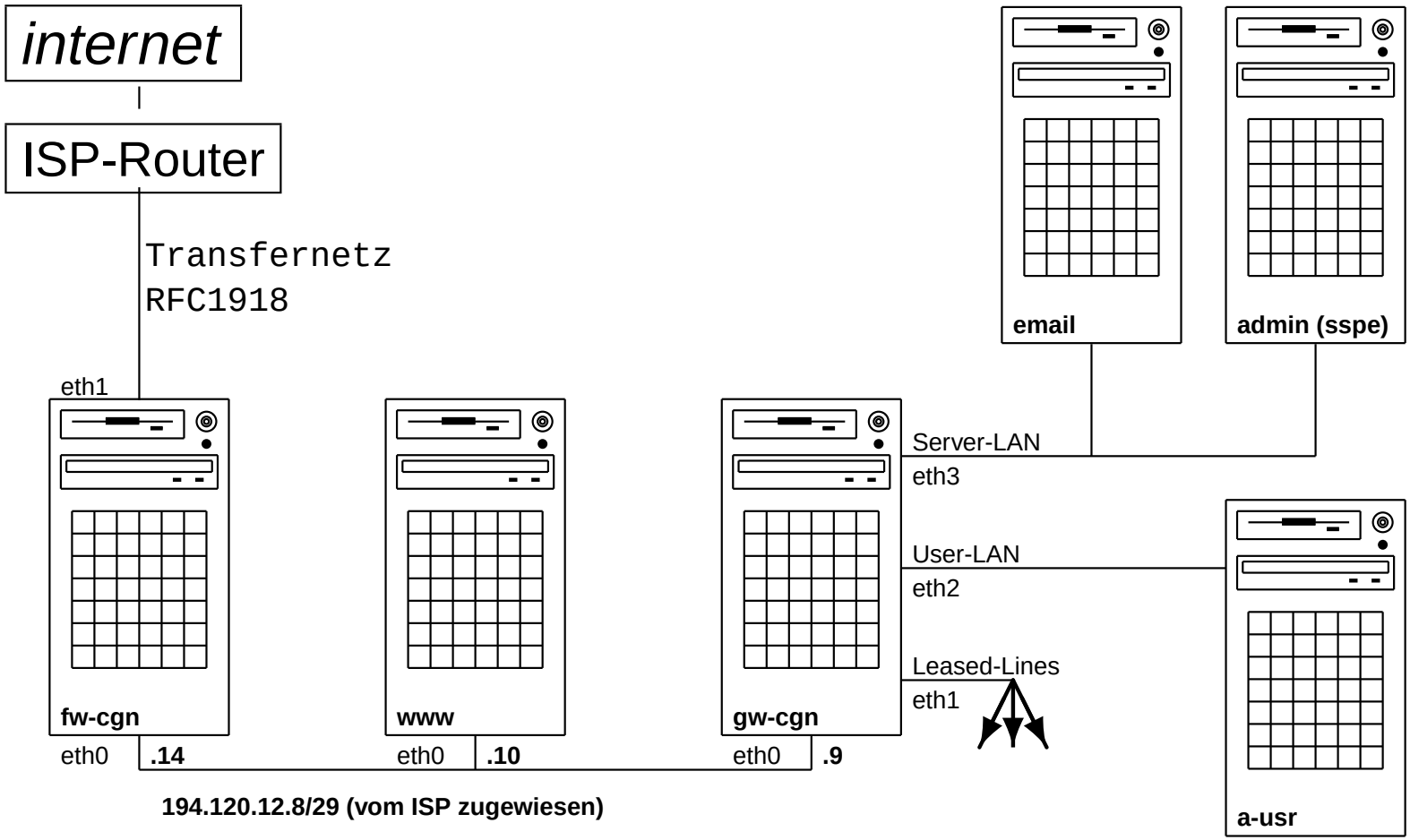
SSPE ist freie Software und unterliegt der
GNU General Public License

Beispielhaft: ein Firmennetzwerk



6 Standorte an beliebigen Internet-Providern

Beispielhaft: typischer Firmenstandort

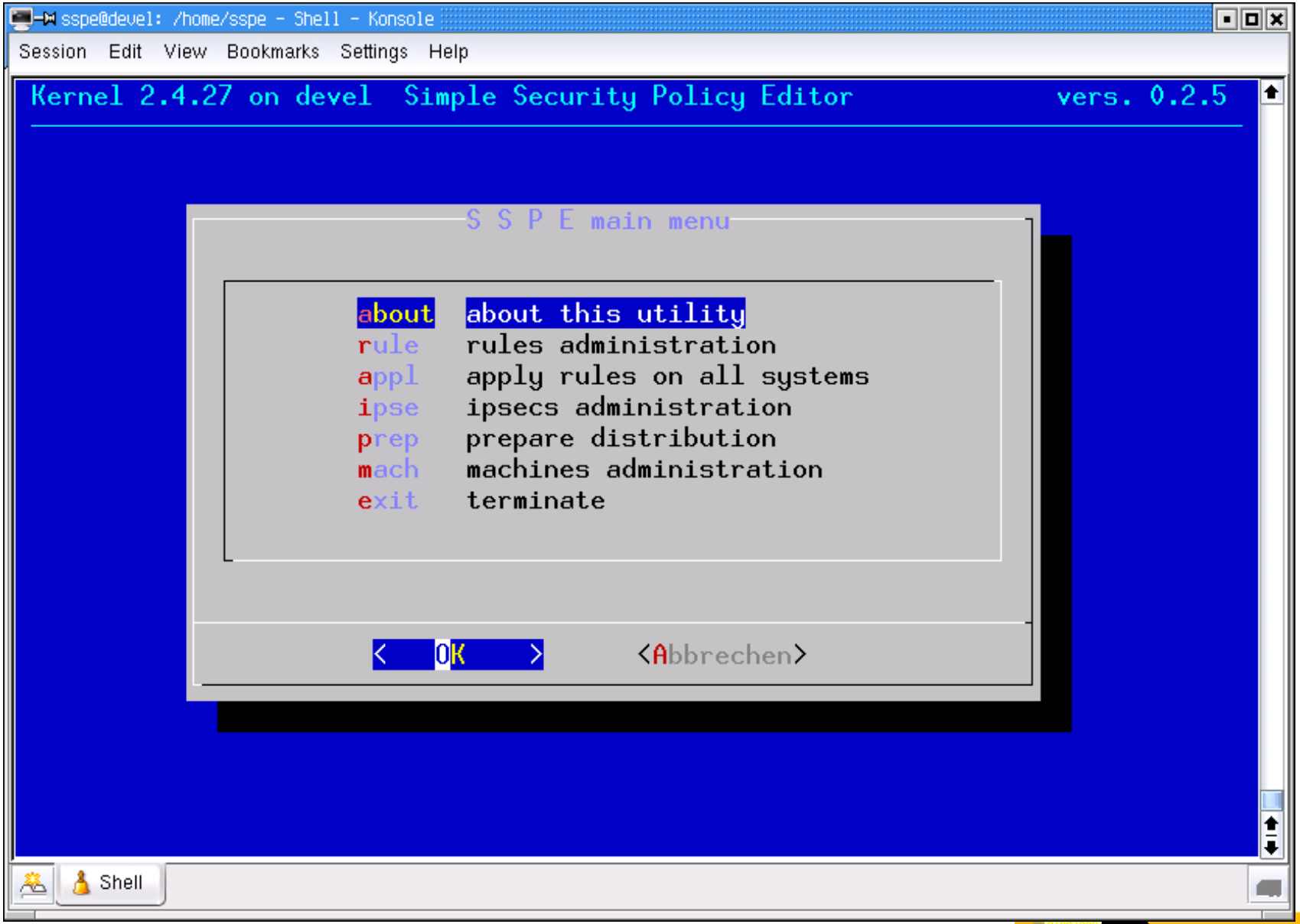


Der Standort des Admin-PC spielt keine Rolle.

Firewall: Entwurfskriterien

- Admins Traum: sowenig Arbeit wie möglich $\Longleftrightarrow$ Faulheit stärkt die Glieder
- zentrale Administration $\Rightarrow$ Konsistenz
- Fehler führen nicht zum Abbruch $\Rightarrow$ Verfügbarkeit
- Top-Down Softwareentwurf
- Inselumgebung für die ersten Versuche
- Linux™ und Cisco™ als erste Plattformen
- Dialog als Rahmen

Firewall: dialog



Hauptmenü

Firewall: hostnet

Definitionen in CIDR-Notation:

```
# File: hostnet
# Name      Address                 # Comment
#
any         0.0.0.0/0               # the whole        internet
many        0.0.0.0/1               # lower half       internet
many        128.0.0.0/1             # upper half       internet
#
a-usr       192.168.1.126/32        # Alice            user-LAN
a-usr       192.168.1.125/32        # Bob              user-LAN
admin       192.168.1.193/32        # sspe-home        server-LAN
gw-cgn      192.168.1.222/32        # gateway cologne  server-LAN
gw-cgn-e    194.120.12.9/32         # gateway cologne  external
cgn-e       194.120.12.8/29         # cologne net      external
fw-cgn      194.120.12.14/32        # firewall cologne external
user-cgn    192.168.1.0/25          # users            user-LAN
cgn-net     192.168.1.0/24          # cgn completely   internal
```

Gruppierung erfolgt durch Namensgleichheit

Firewall: rules

```
# File: rules.admin
# Src      Dst       Dir Prot Port Action Options
#
a-usr      admin     1   tcp  ssh  accept INSEC
many       admin     1   tcp  ssh  deny
admin      gw-cgn    1   tcp  ssh  accept
#
```

```
Dir    =  [ 1 | 2 ]
Prot   =  [ ip | icmp | tcp | udp | esp | 0 \ldots 255 ]
Port   =  [ name | num = 0 \ldots 65535 | :num | num: | num1:num2 ]
Action =  [ accept | reject | deny ]
```

Firewall: Abhängigkeiten

Inhaltliche Abhängigkeiten der generierten Kommandos

- Host-, Netzdefinitionen
- Firewall Regelsatz
- Interfaces, Routingtabelle
- nathosts, privates
- Paketmangling-Dateien

Zeitliche Abhängigkeiten während der Generierung

- sleep-before=2
- wait-before=admin
- Zusammengefaßt in der Datei: apply-options

Firewall: VPN - Implikationen

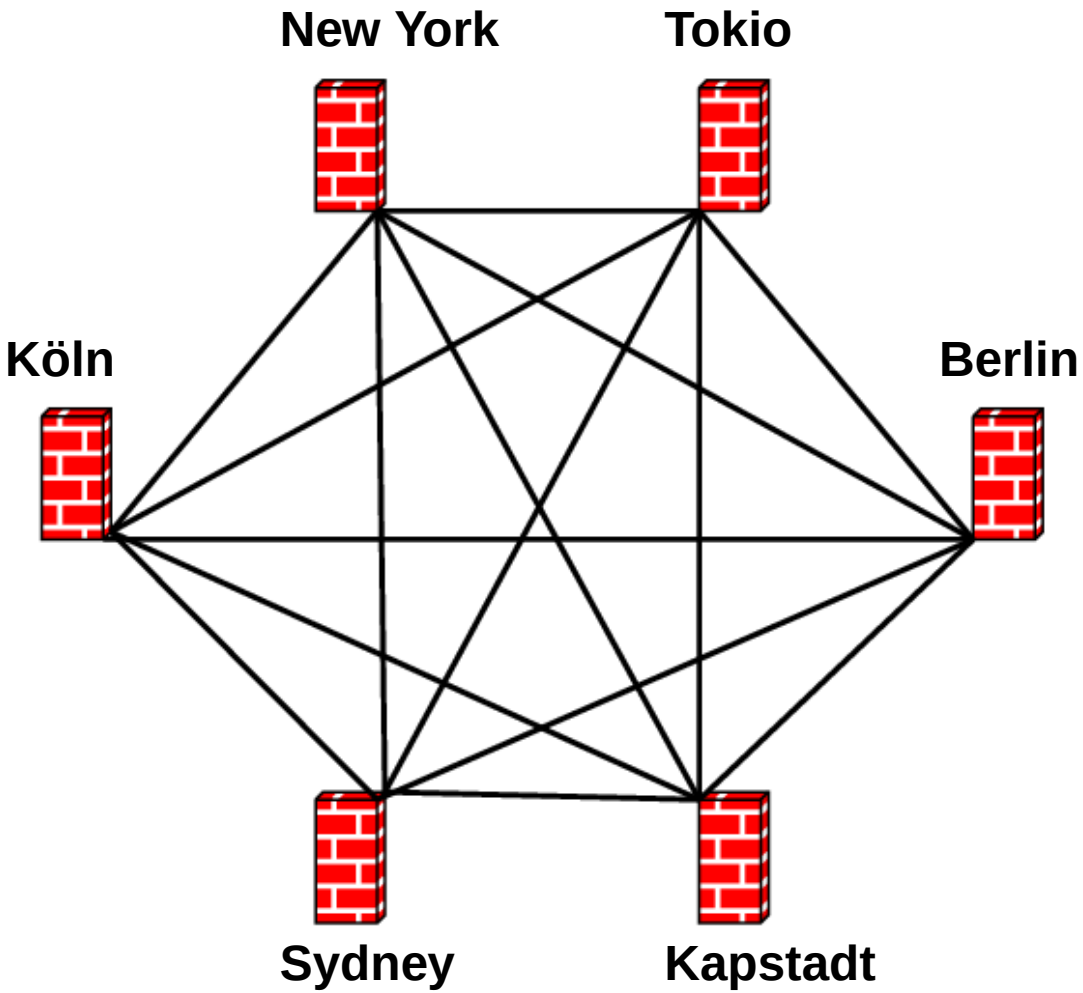
ssh und IPSec

- ausschliesslich ssh zur Administration
- IPSec und ssh nicht wechselseitig abhängig
- ssh durch IPSec nur zu internen Maschinen ohne IPSec
- IPSec verändert Routing, hat also Einfluß auf Generierung!

Wichtigste Erkenntnis:

Paranoid zu sein bedeutet nicht, daß keiner hinter einem her wäre!

VPN: das Firmennetzwerk



6 Standorte an beliebigen Internet-Providern
per IPsec voll vermascht mit $S * (S - 1) = 30$Tunneln

VPN: erster Denkansatz

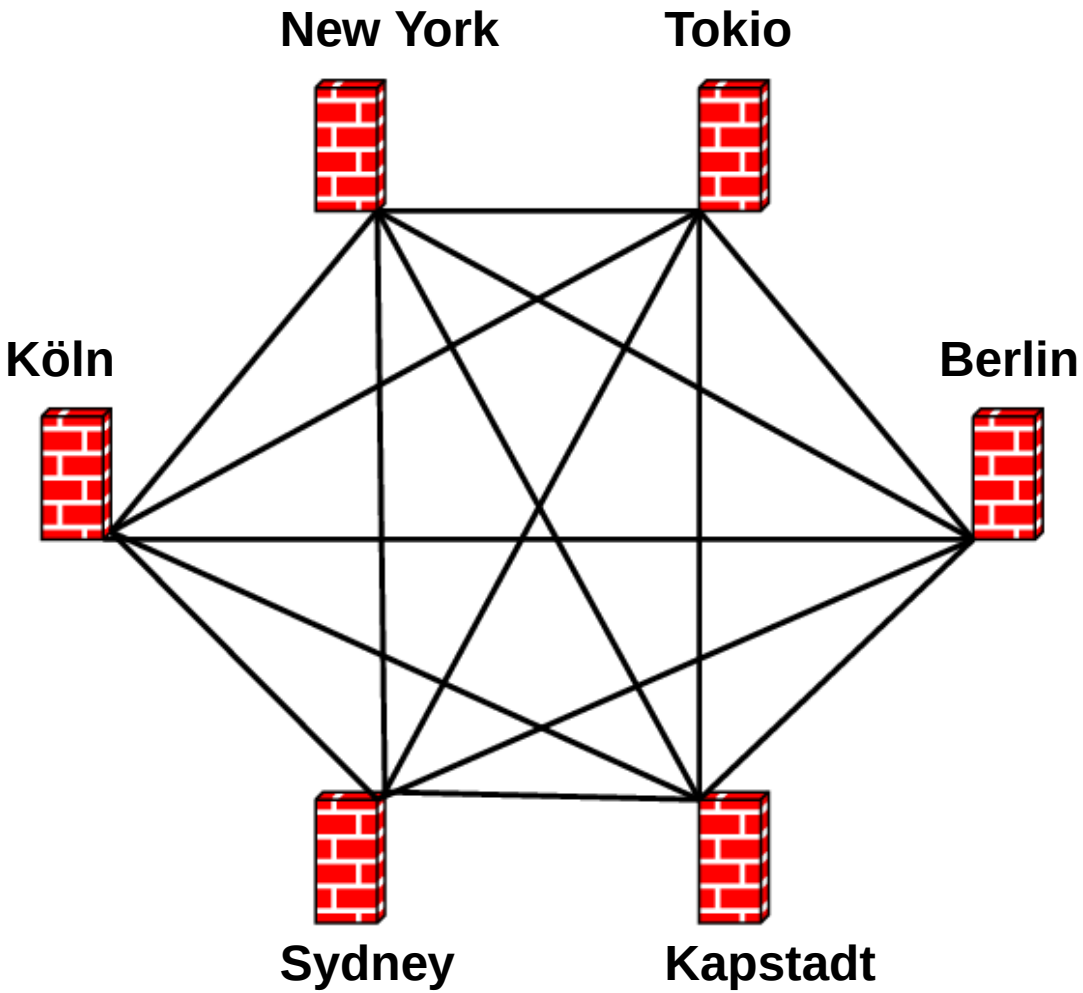
- Gleiche ipsec.conf an allen Standorten, d.h.
  pluto wählt die passenden Verbindungen aus
- Voraussetzung: alle sind gleichzeitig erreichbar
- Zeitsteuerung manuell, Neuladen per cron und ntp synchron sinnvoll
- Overhead für Änderungen ist erträglich,
  30 Sekunden downtime bei der Neukonfiguration
- Konfiguration und PreSharedKeys aus sspe-konfig: ipsecs
- voll vermaschtes Netz, singuläre Standort-Anbindung zusätzlich möglich
- Verteilung mit scp: ipsec.conf.new
- supervisor-script prüft und aktiviert Konfiguration

VPN: ipsecs Konfigurationsdatei

```
# loc.   gateway          next-Hop          subnet
bln      172.22.0.41      172.22.0.46       10.11.48.0/21
cgn      172.22.0.25      172.22.0.30       10.11.40.0/21
nyc      172.22.0.65      172.22.0.70       10.11.4.0/22
sdy      172.22.0.17      172.22.0.22       10.0.0.0/8
kap      172.22.0.9       172.22.0.14       10.11.56.0/21
tok      172.22.0.1       172.22.0.6        10.11.16.0/21
to2      172.22.0.1       172.22.0.6        10.11.80.0/21
```

Hieraus werden alle ipsec.conf und ipsec.secrets generiert

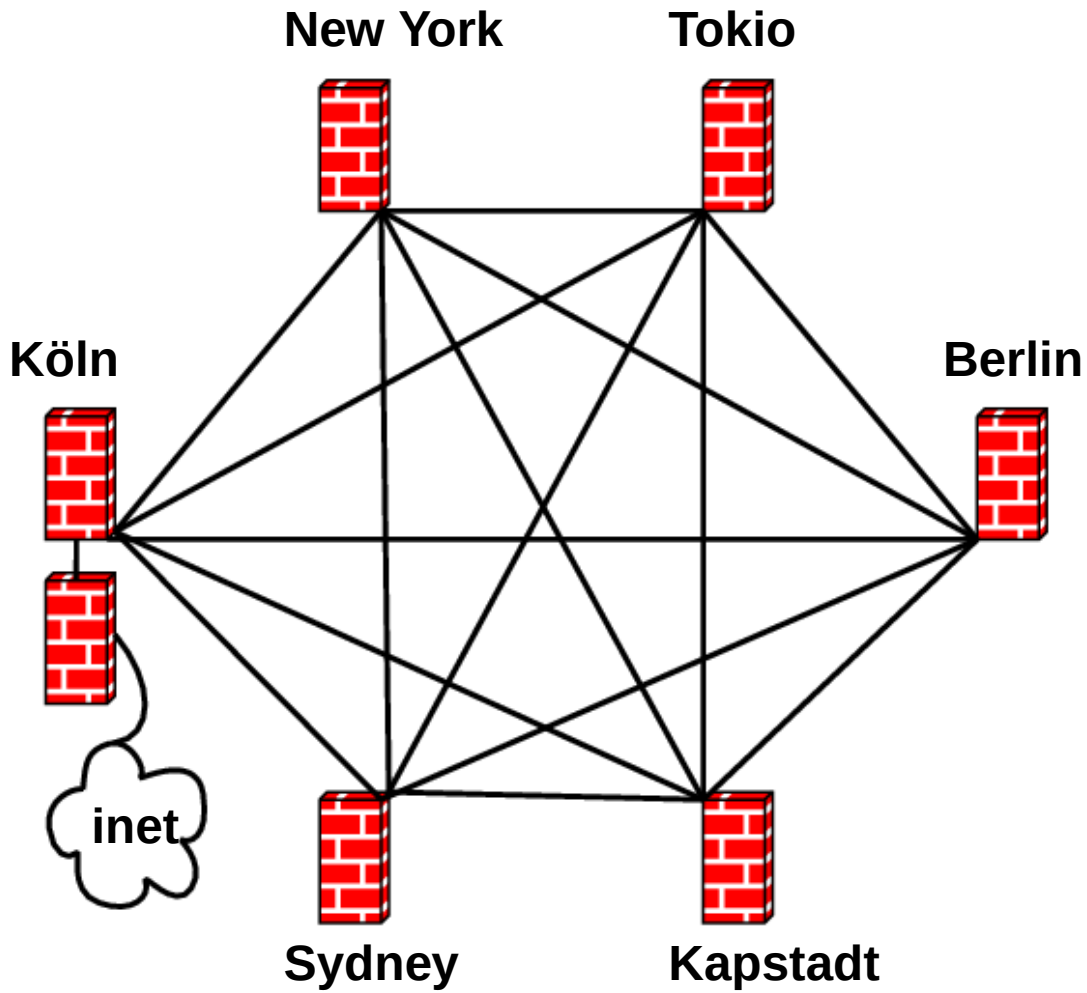
VPN: das Firmennetzwerk vor dem Umbau



6 Standorte an beliebigen Internet-Providern

IPsec voll vermascht mit $S * (S - 1) = 30$Tunneln

VPN: das Firmennetzwerk nach dem Umbau



1 ISP + 6 Standorte an einem ISP-MPLS-VPN,
IPsec voll vermascht mit $(S + 12) * (S - 1) = 90$Tunneln

VPN: ipsecs Konfigurationsdatei

```
# loc.  gateway          next-Hop          subnet
bln     172.22.0.41      172.22.0.46       10.11.48.0/21
cgn     172.22.0.25      172.22.0.30       10.11.40.0/21
nyc     172.22.0.65      172.22.0.70       10.11.4.0/22
sdy     172.22.0.17      172.22.0.22       10.0.0.0/8
kap     172.22.0.9       172.22.0.14       10.11.56.0/21
tok     172.22.0.1       172.22.0.6        10.11.16.0/21
to2     172.22.0.1       172.22.0.6        10.11.80.0/21
I01     172.22.0.25      172.22.0.30       0.0.0.0/1
I02     172.22.0.25      172.22.0.30       128.0.0.0/3
I03     172.22.0.25      172.22.0.30       160.0.0.0/5
I04     172.22.0.25      172.22.0.30       168.0.0.0/6
I05     172.22.0.25      172.22.0.30       172.0.0.0/12
### !!! never open next line or gateways will be lost !!!
### !!!Ixx 172.22.0.25  172.22.0.30       172.16.0.0/12 !!!
I06     172.22.0.25      172.22.0.30       172.32.0.0/11
I07     172.22.0.25      172.22.0.30       172.64.0.0/10
I08     172.22.0.25      172.22.0.30       172.128.0.0/9
I09     172.22.0.25      172.22.0.30       173.0.0.0/8
I10     172.22.0.25      172.22.0.30       174.0.0.0/7
I11     172.22.0.25      172.22.0.30       176.0.0.0/4
I12     172.22.0.25      172.22.0.30       192.0.0.0/3
```

Hieraus werden alle ipsec.conf und ipsec.secrets generiert

VPN: weitere Möglichkeiten

IPsecVPN, etwas kompliziert in der Handhabung

- Handlungsreisende (roadwarrior) mit X.509-Authentisierung
- vpndialer.sf.net für IPsec vom beliebigen M$-PC
  (freie Software von Thomas Kriener)
- Sperrliste für einzelne Clients: CRL der PKI
- L2TP (durch vpndialer initiiert) durch IPsec zur Änderung des Routings im PC

OpenVPN, eine sinnvolle Ergänzung

- Konfiguration einfach, überschaubar und flexibel
- einfacher und weit verbreiteter Client
- PKI kann genutzt werden

Betrieb: Erfahrungen

- Produktionseinsatz seit April 2002
- Mehrerer Kunden und interner Bedarf gedeckt
- einige hundert Anwender-PC geschützt
- Debian stable verdient seinen Namen
- Debian macht security-fixes einfach
- RedHat funktioniert auch, SuSe vermutlich ebenso
- Aussichten: OpenBSD, Solaris, HA, dyn.Routing, . . .
- Kosten drastisch minimiert gegenüber kommerzieller Firewall-Lösung

Dokumentation

Eine Sicherheitsarchitektur ist nur so gut wie ihre Dokumentation
Bei sspe wird sie mit LaTeX aus der laufenden Konfiguration erzeugt:

- Übersicht der Netzwerkarchitektur
  einmalig zu zeichnendes Bild(dia),
  pstricks mit Referenzen (Seitenzahlen der Geräte-Seiten)
- Konfiguration der einzelnen Maschinen
  Interfaces, Routing, ...
- Firewall Definitionen und Regeln
- VPN Konfiguration
- Geplant ist eine weitgehende Vollständigkeit, d.h.
  alles sollte aus der Dokumentation wiederherstellbar sein

Webseite: Mittelstand sicher im Internet

http://www.mittelstand-sicher-im-internet.de/

IT-Sicherheit bei Open-Source-Software

. . . Deshalb setzten z.B. Sicherheitsbehörden in kritischen Bereichen Open-Source-basierte Lösungen ein, deren Vertrauenswürdigkeit zuvor anhand der Quelltexte überprüft wurde. . . .

. . . Die Verantwortung für eine sichere Konfiguration und Wartung der Software bleibt deshalb auch bei Open-Source-Produkten beim Unternehmen. Die Verwendung unsicherer Voreinstellungen, schwache Passwörter und das Betreiben nicht benötigter Dienste auf dem System bleiben – wie bei proprietärer Software auch – eine Gefahr, die nur ein ausgebildeter Administrator eingrenzen kann.

Zusammenfassung

- Freie Programme und ihre Daten – Verfügbarkeit strebt gegen ∞
- kostenloser Quelltext, Kostenvergleich bzgl. Administration entscheidend?
- Quelltext macht Verstehen möglich
- Quelltext macht Änderungen möglich
- Quelltext macht Hintertüren fast unmöglich
- sichere Kryptografie ohne Quelltext ist undenkbar
- Firewalls und VPNs mit freier Software sind möglicherweise sicherer als proprietäre Lösungen
- Firewalladministrator soll verstehen, wie seine Geräte funktionieren
- Nur mit freier Software hat er eine reale Chance!

Danke

Ich bedanke mich für die Aufmerksamkeit bei meinen 44 Folien und wünsche

Frohes Schaffen

Johannes Hubertz