System- und Netzwerksicherheit (IPv4/IPv6) unter Linux

von Dr. Peter Bieringer

Linux Info Tag der Linux User Group Landau
14. Oktober 2006
in Landau/Pfalz, Deutschland

Inhalt

- Über mich
- Typische Gefahren aus dem Internet
- Typische Internet-Anbindungen unter Linux
- Eintrittswahrscheinlichkeiten von Angriffen
- Lösungsansätze zur Vermeidung der Gefahren
- Lösungen im Detail
- Zusammenfassung & Ausblick
- Weitere Informationen

Über mich (bzw. wer bin ich)

- Wohnhaft in München (Deutschland)
- Angestellt bei *AERAsec Network Services and Security GmbH* (seit 2000)
  - Schwerpunkt IT-Sicherheit und Netzwerk-Consulting
  - Trainer für IPv6, TCP/IP und anderes
- Mitgründer und Hauptmitglied von *Deep Space 6*
- Mitglied der Deutschen IPv6 Task Force

Meine Internet-Historie

- 1993: Erster Kontakt mit dem Internet (Univ., SunOS)
- 1996: Erste Erfahrungen mit IPv6 und Linux
- 1997: *IPv6 & Linux - HowTo*, *initscripts-ipv6*
- 1999: *IPv6 & Linux - Current Status*
- 2001: *Linux IPv6 HOWTO*, *ipv6calc*
- 2002: Mitgründer von *Deep Space 6*
- 2006: Mit-Autor Fachbuch *Linux im Netz*

Typische Gefahren aus dem Internet

Typische Gefahren aus dem Internet

- Von außen (Hacker, Würmer, Scanner)
  - Art
    - Angriff auf Server-Applikationen (bereitgestellte Dienste)
      - schwache Passwörter für Authentifizierung (openssh, ftp, webmin)
      - (bekannte) Schwachstellen mit Exploits (Webserver, evtl. mit PHP)
      - ggf. vorher Portscans
      - gezielt (Hacker) oder automatisch (z.B. Wurm)
    - Angriff auf Netzwerk-Stack des Kernels
  - Voraussetzung für erfolgreichen Angriff
    - Unnötige offene Ports, fehlende Zugriffsbeschränkung
    - Schlechtes bzw. fehlendes Firewalling
    - Schwache Passwörter
    - Fehlende Updates für Server-Applikationen und Kernel

Typische Gefahren aus dem Internet

- Von innen (z.B. Trojaner)
  - Art
    - Angriffe auf Client-Applikationen
      - E-Mail: Outlook (Windows), Thunderbird
      - Web-Browser: Internet Explorer mit ActiveX (Windows), Firefox, Flash-Plugin
      - Applikationen: Office, Adobe Reader (JavaScript)
    - Angriffe auf das Vertrauen des Benutzers
      - Phishing, Pharming
  - Voraussetzung für erfolgreichen Angriff
    - Fehlendes Mißtrauen („Mithilfe“) des Benutzers
    - Fehlende Updates für Client-Applikationen

Typische Internet-Anbindungen unter Linux

Typische Internet-Anbindungen unter Linux

- Direkte Wählverbindung
  - Analog/ISDN/(A)DSL-Modem
  - ISDN-Karte
- Lokales Netzwerk mit Internet-Anbindung
  - mit Hilfe eines Analog/ISDN/(A)DSL-Router

Direkte Wählverbindung

- Realisierung

mit Hilfe von	Schnittstelle	Übertragung
Analog-Modem	ppp+	PPP über seriell
(A)DSL-Modem	ppp+	PPP über Ethernet (PPPoE)
ISDN-Modem	ppp+	PPP über seriell / HDLC
ISDN-Karte	ippp+	PPP über HDLC

- Zuweisung globaler IP-Adresse

  erfolgt direkt an eine (logische) Schnittstelle im PC!

  - IPv4-Adresse: vom Internet Service Provider (ISP)
  - ggf. IPv6-Adresse: meist vom Tunnel-Provider

Lokales Netzwerk mit Internet-Anbindung

- Realisierung

mit Hilfe von	Schnittstelle	Übertragung
Ethernet-Schnittstelle	eth+	Ethernet

- Zuweisung globaler IP-Adresse
  - IPv4-Adresse: keine (üblicherweise nur eine lokal gültige)
    - ggf. aber Port-Forwarding, z.B. für Fernwartung
    - ggf. Verbindung durch STUN möglich
  - ggf. IPv6-Adresse: nach Erhalt eines Präfix vom Router

Eintrittswahrscheinlichkeiten von Angriffen

Eintrittswahrscheinlichkeiten

Protokoll:	IPv4			IPv6
Verbindung:	direkte Wählverbindung	Lokales Netzwerk kein Port-Forwarding	Lokales Netzwerk Port-Forwarding	nativ
Angriff auf:				
Netzwerk-Stack	+	-	o	++
Server-Applikationen	+++	--	++	+
Client-Applikationen	+	+	+	++
Benutzer-Vertrauen	+++	+++	+++	o

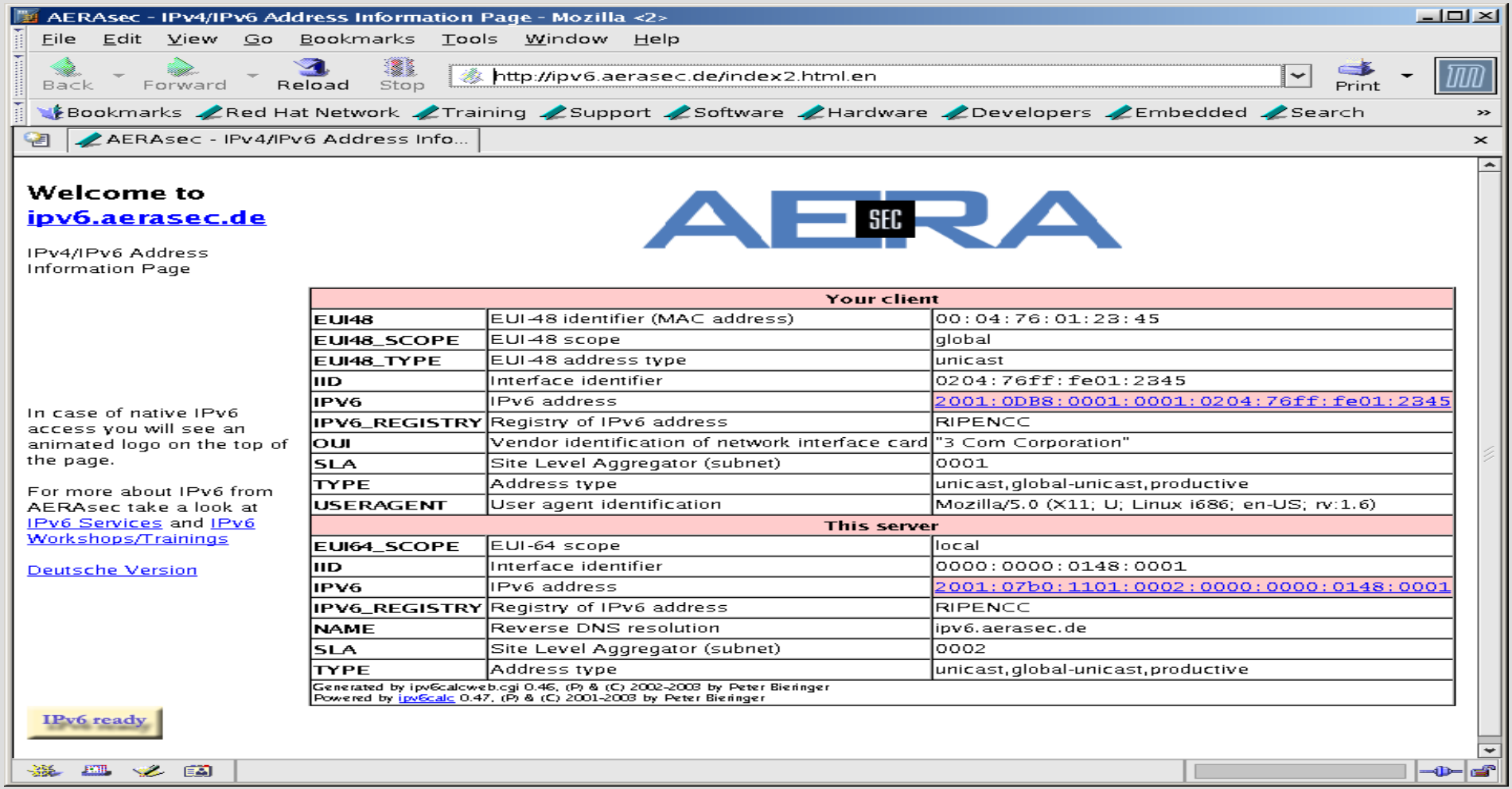
Firewalling beim Einsatz von IPv6 ist sehr wichtig!

Es gibt keinen impliziten Schutz mehr - im Gegensatz zu IPv4 mit Hilfe von NAT!

Test auf funktionsfähige IPv6-Verbindung

```
# ping6 ipv6.aerasec.de
```

URL: http://ipv6.aerasec.de/



Your client		
EUI48	EUI-48 identifier (MAC address)	00:04:76:01:23:45
EUI48_SCOPE	EUI-48 scope	global
EUI48_TYPE	EUI-48 address type	unicast
IID	Interface identifier	0204:76ff:fe01:2345
IPV6	IPv6 address	2001:0DB8:0001:0001:0204:76ff:fe01:2345
IPV6_REGISTRY	Registry of IPv6 address	RIPENCC
OUI	Vendor identification of network interface card	"3 Com Corporation"
SLA	Site Level Aggregator (subnet)	0001
TYPE	Address type	unicast,global-unicast,productive
USERAGENT	User agent identification	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6)
This server		
EUI64_SCOPE	EUI-64 scope	local
IID	Interface identifier	0000:0000:0148:0001
IPV6	IPv6 address	2001:07b0:1101:0002:0000:0000:0148:0001
IPV6_REGISTRY	Registry of IPv6 address	RIPENCC
NAME	Reverse DNS resolution	ipv6.aerasec.de
SLA	Site Level Aggregator (subnet)	0002
TYPE	Address type	unicast,global-unicast,productive

Generated by ipv6calcweb.cgi 0.46, (P) & (C) 2002-2003 by Peter Bieringer
Powered by ipv6calc 0.47, (P) & (C) 2001-2003 by Peter Bieringer

Lösungsansätze zur Vermeidung der Gefahren

Lösungsansätze

- Fehlende Updates für Applikationen
  - Regelmäßig (täglich!) auf Updates prüfen
    - Automatische Updates aktivieren
    - Kernel-Update nicht vergessen!
      - Sicherheitsaspekte sprechen gegen lange Uptime!
- Unnötige offene Ports, fehlende Zugriffsbeschränkung
  - Unnötige Dienste abschalten
  - Konfiguration der Dienste verbessern
    - Binden an localhost (127.0.0.1, ::1) bzw. interne IP-Adressen
    - Zugriff beschränken auf localhost bzw. interne IP-Adressen

Lösungsansätze

- Schlechtes bzw. fehlendes Firewalling
  - Lokales Firewalling aktiveren
  - Firewalling auf Router aktivieren/verbessern
- Schwache Passwörter
  - Bessere Passwörter verwenden
  - Zertifikate verwenden
    - openssh: public/private key
    - SSL: Einsatz von Client-Zertifikaten

    Achtung: gute Passphrase für den Schutz des privaten Schlüssels verwenden!

Lösungen im Detail

Einbinden von zusätzlichen Repositories

- Durch Einbinden von zusätzlichen Repositories entfällt die Notwendigkeit, selbst zu kompilieren und zu installieren
- Aktivierung
  - Fedora Core:
    - Zusätzliche <name>.repo-Datei in /etc/yum.repos.d/ ablegen
- Prüfung, ob Paket in einem Repository enthalten ist:
  - Yum-basierende Distributionen:

```
# yum list <paketname>
```

Regelmäßige automatische Updates

- Automatisches Update wird von allen Distributionen unterstützt
  - Pakete werden vom Hersteller bzw. der Community gepflegt
    - Allerdings meist nur für aktuelle Distributionen
- Aktivierung
  - Yum-basierende Distributionen:

```
# chkconfig yum on; service yum start
```

    “yum update” wird dann über cron (bei 24x7 Betrieb) bzw. anacron (unregelmäßige Benutzung des PCs) gestartet
- Vorher Prüfung auf Funktionsfähigkeit durchführen!
  - Yum-basierende Distributionen:

```
# yum update
```

Unnötige offene Ports

- Oft unnötige offene Ports durch Standardinstallation
- Prüfung (als Benutzer „root“)

```
# netstat -nlptu
# lsof -n -i | grep -v '\->'
# rpcinfo -p
```

- Unnötige Dienste abschalten
  - Fedora Core / Red Hat Enterprise Linux:

```
# chkconfig <dienst> off; service <dienst> stop
```

Unnötige offene Ports

- Fedora Core 6 Test 3

```
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:2208          0.0.0.0:*               LISTEN      1788/hpiod
tcp        0      0 127.0.0.1:58371         0.0.0.0:*               LISTEN      1793/python
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1601/portmap
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1803/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1830/sendmail: acce
tcp        0      0 0.0.0.0:954             0.0.0.0:*               LISTEN      1620/rpc.statd
tcp        0      0 :::22                   :::*                    LISTEN      1812/sshd
tcp        0      0 ::1:631                 :::*                    LISTEN      1803/cupsd
udp        0      0 0.0.0.0:32768           0.0.0.0:*                           1930/avahi-daemon:
udp        0      0 0.0.0.0:948             0.0.0.0:*                           1620/rpc.statd
udp        0      0 0.0.0.0:951             0.0.0.0:*                           1620/rpc.statd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1486/dhclient
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           1930/avahi-daemon:
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1601/portmap
udp        0      0 0.0.0.0:631             0.0.0.0:*                           1803/cupsd
udp        0      0 :::32769                :::*                                1930/avahi-daemon:
udp        0      0 :::5353                 :::*                                1930/avahi-daemon:
```

⇨ **Absicherung notwendig bzw. prüfen**

Prio1: SSH-Port (22)

Prio2: Portmapper (111) und RPC-Dienste

Unnötige offene Ports

- Ubuntu Dapper Drake

```
Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:34759         0.0.0.0:*               LISTEN      4059/python
tcp        0      0 127.0.0.1:45907         0.0.0.0:*               LISTEN      4056/hpiod
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      4107/cupsd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           2976/dhclient3
```

⇨ **Keine weitere Absicherung notwendig**

Binden an localhost bzw. an interne IP-Adressen

- Oft sind Dienste an „any“ (0.0.0.0 bzw. ::) gebunden, obwohl nur lokal bzw. intern genutzt
- Prüfung (als Benutzer „root“)

```
# netstat -nlptu | egrep " 0.0.0.0:[^*]| :::[^*]"
# lsof -n -i | grep " \*:"
```

- Konfiguration der Dienste anpassen (Beispiele)

Squid [/etc/squid/squid.conf]:

```
http_port 192.168.1.1:3128
http_port 127.0.0.1:3128
```

Samba [/etc/samba/smb.conf]:

```
interfaces = 192.168.1.1/24 127.0.0.1/8
bind interfaces only = Yes
```

Zugriffsbeschränkung auf localhost bzw. auf interne IP-Adressen

- Oft erlauben Server-Applikationen den Zugriff von überall her
- Prüfung
  - Verbindungsaufbau von außen prüfen
- Konfiguration der Dienste anpassen (Beispiele)

Squid [/etc/squid/squid.conf]:

```
acl localnet src 192.168.1.0/255.255.255.0
http_access allow localnet
http_access deny all
```

cups [/etc/cups/cupsd.conf]:

```
Deny From All
Allow From 127.0.0.1
Allow From 192.168.1.0/24
```

Zugriffsbeschränkung auf localhost bzw. auf interne IP-Adressen

- Zur weiteren (zusätzlichen) Absicherung tcp_wrapper konfigurieren
  - wichtig für Dienste, die weder Änderung des Binden der IP-Adresse noch eine Zugriffsbeschränkung direkt erlauben
- Prüfung:
  - Verbindungsaufbau von außen prüfen
- Datei /etc/hosts.allow anpassen (Beispiele)

  portmap:

  ```
  portmap: 192.168.1. 127.0.0.1
  ```

  openssh:

  ```
  sshd: 192.168.1. 127.0.0.1 .dip.t-dialin.net
  ```

Verstellen des Server-Ports

- Wenn der Server-Port vom well-known-Port auf einen anderen (höheren) verschoben wird, liegt er außerhalb der einfachen automatischen Angriffen
  - Zur Zeit üblich (Linux-betreffend):
    - 21 (ftp) 22 (ssh) 25 (smtp)
    - 80 (http) 5900 (vnc) 10000 (webmin)
- Prüfung (als Benutzer „root“)

```
# netstat -nlptu
# lsof -n -i | grep -v '\->'
```

- Konfiguration der Dienste anpassen (Beispiele)

openssh (well-known-Port 22) [/etc/ssh/sshd_config]:

```
Port 12322
```

Linux-Firewalling allgemein

- Paketfilter in aktuellen Linux-Kernel heißt „netfilter“
  - ersetzte 2001 ipchains ab Kernel 2.3.x
  - Benutzer-Werkzeug: *iptables* (IPv4) bzw. *ip6tables* (IPv6)
- Filter-Möglichkeiten
  - IPv4: abhängig von den einkompilierten Modulen gibt es diverse “Helper” für kompliziertere Protokolle (wg. NAT und *connection tracking*)
  - IPv6: aktuell noch kein *connection tracking* möglich, Entwicklung aber bereits weit fortgeschritten

Lokales Firewalling

- Lokales Firewalling schützt den einzelnen PC vor Angriffen über das Netzwerk (intern und von außen)
- Konfiguration des Firewallings
  - IPv4:
    - distributionseigene Werkzeuge
      - Fedora Core: system-config-securitylevel (ab FC6 auch für IPv6)
    - *fwbuilder*
  - IPv6:
    - Aufbau eigenes Regelwerks
      - Hilfsskripte: ip-firewalling (unterstützt IPv4 und IPv6)
        ftp://ftp.aerasec.de/pub/linux/repository/public/redhat/enterprise/4/i386/

Firewalling auf Access-Router

- Kommerzielle Access-Router (DSL)
  - Gängige Produkte können nur IPv4 filtern
  - meist wird IPv6 gar nicht unterstützt, somit steht intern keine IPv6-Anbindung zur Verfügung
    - Ausnahme: IPv6 über UDP-Tunnel
- Linux-Router im Eigenbau
  - Können IPv6 unterstützen
    - Linux-Distributions-basiert
    - openwrt

Beispiel für IPv6 netfilter Regel

- Minimales Regelwerk (für *ip6tables-restore*):

```
*filter
:INPUT DROP [12:440]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [12:440]
-A INPUT -s ::/0 -d ::/0 -p tcp -m tcp --dport 512:65535 ! --tcp-flags
  SYN,RST,ACK SYN -j ACCEPT
-A INPUT -s ::/0 -d ::/0 -p udp -m udp --dport 512:65535 -j ACCEPT
-A INPUT -p icmpv6 -j ACCEPT
-A INPUT -m limit --limit 5/min -j LOG
-A INPUT -j DROP
COMMIT
```

- Erlauben eingehend SSH von überall her:

```
-A INPUT -p tcp --dport 22 -j ACCEPT
```

- Mehr Tipps stehen zur Verfügung unter:

  http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/chapter-firewalling-security.html

Schwache Passwörter

- Schwache Passwörter bei Server-Applikationen an well-know-Ports werden automatisiert mit Wörterbuch-Attacken angegriffen
- Prüfung:
  - Passwörter prüfen
  - Ggf. Einschränken der Benutzer
- Konfiguration der Dienste anpassen (Beispiele)

  openssh [/etc/ssh/sshd_config]:

```
AllowUsers peter
AllowGroups wheel
```

Einsatz von Zertifikaten

- Passwörter entsprechen der 1-Faktor-Authentifizierung („what you know“)
- Durch den Einsatz von Zertifikaten kann eine 2-Faktor-Authentifizierung ermöglicht werden („what you have“ & „what you know“)
- Konfiguration der Dienste anpassen (Beispiele)

  openssh (nach Erstellung der Schlüssel)

  [/etc/ssh/sshd_config]:

```
PasswordAuthentication no
```

Zusammenfassung & Ausblick

Zusammenfassung

- Systemsicherheit eines Linux-Systems ist machbar
  - Einsatz einer aktuellen Distribution
  - Regelmäßige (automatische) Updates
  - Geringer Aufwand!
- Netzwerksicherheit eines Linux-Systems ist machbar
  - Hoher Schutz durch Kopplung möglich
    - Abschalten unnötiger Dienste
    - Binden der Dienste auf localhost oder internes Netz
    - Dienst-eigenen Zugriffschutz konfigurieren
    - tcp_wrapper konfigurieren
    - Firewalling konfigurieren

Ausblick

- Systemsicherheit eines Linux-Systems
  - Bedrohungslage bleibt bestehen, Update-Zyklen werden schneller
  - Schadensbegrenzung durch Mandatory Access Control (SELinux, AppArmor, RSBAC)
- Netzwerksicherheit eines Linux-Systems
  - Lokales Firewalling ist bei Einsatz von IPv6 Pflicht!
    - Erweiterter Schutz auch durch ausgehendes Firewalling
    - Connection Tracking für IPv6 (hoffentlich) bald verfügbar

Weitere Informationen

Linux bezogene Information

- *Netfilter* *http://www.netfilter.org/*
- *Firewall Builder* *http://www.fwbuilder.org/*
- *NMAP* *http://insecure.org/nmap/*
- *Nessus* *http://www.nessus.org/*
- *http://www.aerasec.de/security/systems/linux.html*
- *Checks der c't*
  - *Netzwerk: http://www.heise.de/security/dienste/portscan/*
  - *Browser: http://www.heise.de/security/dienste/browsercheck/*

IPv6 & Linux bezogene Information

- *Linux IPv6 HOWTO*
  - Schwerpunkt: ausgiebige Information über IPv6 in Linux
    http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/ (nur English)
    http://mirrors.bieringer.de/ (en, de, fr, it) 
  - URLs zu allen Übersetzungen und weiterführende Informationen
    http://www.bieringer.de/linux/IPv6/
- *Current Status of IPv6 Support for Networking Applications*
  - IPv6-Status von netzwerkfähigen Applikationen
    http://www.deepspace6.net/docs/ipv6_status_page_apps.html
- IPv6-Tunnel Broker (Auswahl): http://www.sixxs.net/
- Teredo/Miredo (IPv6 über UDP):

  http://www.simphalempin.com/dev/miredo/

Kontakt-Informationen

pb@bieringer.de

http://www.bieringer.de/pb/

http://www.bieringer.de/linux/IPv6/

peter@deepspace6.net

http://www.deepspace6.net/

pbieringer@aerasec.de

http://www.aerasec.de/

@OpenBC: http://www.openbc.com/hp/Peter_Bieringer/

Vielen Dank für Ihre Aufmerksamkeit!

Fragen & Antworten

Dankeschön an
Jean-Jacques Sarton (Einladung)