Rechnersicherheit

Situation: Linux-Rechner im Studentenwohnheim am Internet bzw. per PPP (Modem/ISDN) an die Uni angebunden.

Ein Rechner zu dem physikalischer Zugriff besteht, ist nicht sicher! Einzige Abhilfe: Verschlüsslung der Daten, die auf dem Rechner liegen.

Siehe dazu **cfs** (Cryptographic File System), **tcfs** (Transparent Cryptographic File System) bzw. **des** oder **pgp** (Pretty Good Privacy).

Dieser Vortrag beschäftigt sich nur mit Komprimitierung der Rechnersicherheit durch Netzanschluß.

Motivation

Möglicher Schaden durch Vernachlässigung von Sicherheitsaspekten:

- Daten werden Unbefugten zugänglich (lesbar)
- Daten durch Unbefugten verändert, gelöscht
- finanzieller Schaden durch Mißbrauch des Modems
- unberechtigter Zugang zu anderen Rechnern, z. B. Uni
- unbefugtes Lesen von E-Mail, Fälschen von E-Mail
- unsichere Datenübertragung im WWW, “Gläserner Surfer”

Potentielle Sicherheitslücken eines vernetzten Rechners

1. Benutzer erlangt mehr als die zugewiesenen Rechte
   (a) Ausnutzen von Sicherheitslücken (z. B. mount, sendmail)
   (b) Benutzung abgehörter Paßwörter
   (c) Cracken von Paßwörtern

2. Unbefugtes Eindringen in das System
   (a) Ausnutzen von Sicherheitslücken (z. B. telnetd, rexecd)
   (b) Benutzung abgehörter Paßwörter
   (c) Konfigurationsfehler (z. B. ftpd, hosts.equiv, NIS, NFS)
   (d) Sicherheitslücken durch leichsinnige Benutzer
   (e) “Session Hijacking”

3. “Denial of Service Attack”
   (a) Ausnutzen von Sicherheitslücken (z. B. ping bug)
   (b) Konfigurationsfehler (z. B. ftpd)
   (c) “IP Flooding”

Vorgehensweise

1. Klar werden über eigene “Sicherheitspolitik”
2. Schließen der Sicherheitslücken
3. System sicherer machen
4. Erkennen von Angriffen

Schließen der Sicherheitslücken

- Schließen der Sicherheitslücken im System: Lesen der Security-Advisories, Einspielen von Updates
- Konfigurationsfehler suchen z. B. mit SATAN (Security Administrator Tool for Analyzing Networks)
- Sicherheitslücken durch leichsinnige Benutzer schließen: z. B. mit Crack, COPS.

System sicherer machen

- Shadow Paßwörter gegen Paßwortcracker
- tcpd: Zugriff beschränken (auf Applikationsebene)
- Firewall: Zugriff beschränken (auf Paketebene)
- S/Key: Einmalpaßwörter
- Ssh: Übertragung verschlüsseln (auch Paßwörter)

TCP-Wrapper

Der tcpd ermöglicht erweitertes Mitprotokollieren und Zugriffsbeschränkung für Dienste, die vom `inetd` aufgerufen werden.

Beispielausschnitt aus `/etc/inetd.conf`:

```
ftp     stream tcp  nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet  stream tcp  nowait  root    /usr/sbin/tcpd  in.telnetd
finger  stream tcp  nowait  nobody  /usr/sbin/tcpd  in.fingerd
```

tcpd

Zugangsbeschränkung durch `hosts.allow` und `hosts.deny`:

`/etc/hosts.deny`:

```
ALL: ALL
```

`/etc/hosts.allow`:

```
telnetd: .wohnheim.uni-kl.de
rlogind: .wohnheim.uni-kl.de
ftpd: ALL
```

Kann auch vor “Denial of Service Attack” schützen.

Secure Shell (Ssh)

Ssh ersetzt die ‘r’-Programme **rlogin**, **rcp**, **rsh** und bietet dabei starke Authentifizierung und sichere Kommunikation über unsichere Kanäle.

- **slogin** ermöglicht Einloggen übers Netz auf anderen Rechner.
- Mit **scp** können Dateien zwischen vernetzen Maschinen kopiert werden.
- Kommandos können auf anderen Rechnern über das Netzt mit **Ssh** ausgeführt werden.

Ssh bietet weiterhin sichere X-Verbindungen, sowie Forwarding beliebiger tcp-Verbindungen (z. B. FTP oder POP).

Ssh – Verschlüsselte Übertragung

- Übertragung vollständig verschlüsselt – keine Übertragung von Klartextpaßwörtern.
- Benutzung transparent für den Benutzer. Falls auf ander Seite kein **sshd**, Rückgriff auf ‘r’-Programme (nach Warnung).
- Rechenzeitverbrauch durch Verschlüsslung fällt auf modernen Rechnern nicht ins Gewicht. Durch Kompression wird Übertragung über das Netz z. T. schneller
- Ssh schützt auch vor IP-Spoofing (Übernehmen bestehender Verbindungen).

Ssh – Beispiele

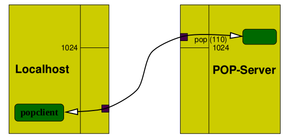
POP und FTP via Ssh (Vorraussetzung, daß man sich per Ssh dem POP/FTP-Server einloggen kann):

```
# Einloggen auf dem POP-Server und POP forwarden
ssh -L 8110:localhost:110 popserver

# Mail per POP holen:
popclient -port 8110 -host localhost
```

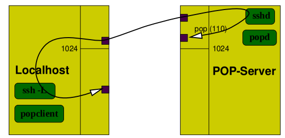
```
# Einloggen auf dem FTP-Server und FTP forwarden
ssh -L 4711:localhost:21 ftp-host

# FTP-Verbindung zum ftp-host aufbauen
ftp localhost 4711
```

Ssh – Diagramm

Normal POP:



Secure POP:



Firewall – Funktionsweise

Mögliche Policies:

1. Was nicht explizit verboten ist, ist erlaubt.

2. Was nicht explizit erlaubt ist, ist verboten.

Abwägen von Sicherheit und bequemer Benutzung des Systems.

- arbeiten auf IP-Paket-Basis, Durchlassen der Pakte je nach Quelle, Ziel, Port oder Inhalt
- oder auf Applikationsebene: WWW-Proxy,FTP-Proxy, Telnet-Proxy …

Linux als Firewall

- Der Linux-Kernel kann als Firewall benutzt werden (compiletime Option: `CONFIG_FIREWALL=y`).
- Filterregeln können zur Laufzeit mit `ipfwadm` eingestellt werden.

```
# default policies:
/sbin/ipfwadm -O -p accept
/sbin/ipfwadm -I -p reject

# eigener Rechner darf alles
/sbin/ipfwadm -I -P all -a accept -S ${HOSTNAME}
/sbin/ipfwadm -I -P all -a accept -S 127.0.0.0/8

# DNS: Nameserver udp von domain zu nicht priv. Ports
/sbin/ipfwadm -I -P udp -a accept -S 131.246.9.116 53\
                                  -D 0.0.0.0/0 1024:65535
```


Erkennen von Angriffen

- syslogd: Mitloggen sicherheitsrelevanter Information
- tcpd: Mitloggen von Dienst, Clients, User oder sonsitigem (Konfigurierbar)
- chklogs: Logfiles automatisiert auswerten
- firewall: Pakete mitloggen

Syslogd – Mitloggen wichtiger Ereignisse

`/etc/syslog.conf:`

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                        /var/log/secure

# Log all the mail messages in one place.
mail.*                                            /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                           *
*.emerg                                           @securehost
```


Potentielle Sicherheitslücken eines Netzes (1)

1. Unbefugtes Eindringen in anderes System (z. B. Uni)
   (a) Benutzung abgehörter Paßwörter → Ssh, S/Key
   (b) "Session Hijacking" → Ssh, tcpd
2. Sicherheit bei der Benutzung von E-Mail
   (a) Mitlesen durch Unbefugte → PGP (Pretty Good Privacy) Verschlüsslung
   (b) E-Mail mit gefälschtem Absender → PGP elektronische Unterschrift

Potentielle Sicherheitslücken eines Netzes (2)

3. Sicherheit bei der Benutzung von WWW
   (a) Sicherheit der Datenübertragung (z. B. Kreditkartendaten) → **SSL** (Secure Socket Layer)
   (b) Erstellung von Benutzerprofilen z. B. durch Cookies, Javascript.
   (c) Verlust der Anonymität bei 1:1 Abbildung von IP-Adresse auf Benutzer.

Literatur / weitere Informationen

Mailingliste: `win-sec-ssc@cert.dfn.de`

URLs:

Sicherheit im DFN: `http://www.cert.dfn.de/`

Linux Security `http://www.redhat.com/linux-info/security/`

SSH-Homepage `http://www.cs.hut.fi/ssh/`

SSH-FAQ `http://www.uni-karlsruhe.de/ ig25/ssh-faq/`

International Cryptography Pages `http://www.cs.hut.fi/crypto/`

Security and Encryption-related Resources and Links:

`http://www.cs.auckland.ac.nz/ pgut001/links.html`

PGP Public-Keyserver `http://www-math.uni-paderborn.de/pgp/`

Newsgruppen:

- `comp.security.announce`
- `de.comp.security`

Bücher: (O'Reilly `http://www.ora.de/`)

- O'Reilly: TCP/IP Netzwerk Administration (Craig Hunt)
- O'Reilly: Einrichten von Internet-Firewalls (Brent Chapman und Elizabeth D.Zwicky)
- O'Reilly: PGP: Pretty Good Privacy (Simson Garfinkel)
- O'Reilly: Satan Sicherheitsmängel erkennen und beheben (Martin Freiss)