TCP-Wrapper

- Kurzakte -

Ursprünglich alleine für die Verwendung mit [x]inetd gedacht, hat sich das Prinzip des TCP-Wrappers inzwischen auch für Stand-Alone-Dienste durchgesetzt:

- SSH
- NIS

Dazu wird einfach statt dem zu startenden Dienst der tcpd aufgerufen und ihm wird der Name des zu startenden Dienstes als Parameter mitgegeben. Das Programm tcpd überprüft jetzt anhand von Einträgen in den Dateien

- /etc/hosts.allow
- /etc/hosts.deny

ob der Dienst von dem entsprechenden Host in Anspruch genommen werden darf. Analog überprüfen auch Stand-Alone-Dienste diese beiden Dateien.

inetd und TCP-Wrappers

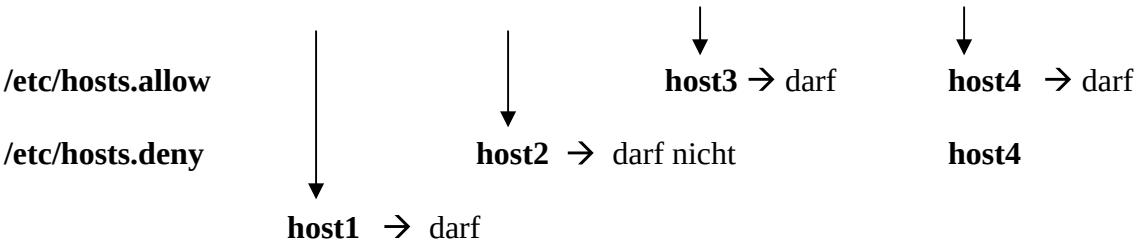
- inetd kann Dienste über den tcpd aufrufen und somit die TCP-Wrapper abarbeiten.

xinetd und TXP-Wrappers

- xinetd benötigt nicht den tcpd, sondern bietet eine fest eingebaute Unterstützung der Abarbeitung der Dateien /etc/hosts.allow und /etc/hosts.deny.

Die Überprüfung erfolgt auf eine etwas eigenwillige Weise:

- Existiert ein passender Eintrag in der Datei /etc/hosts.allow, so wird Zugriff gegeben. Wenn nicht, dann
- Existiert ein passender Eintrag in der Datei /etc/hosts.deny, so wird kein Zugriff gegeben. Wenn nicht, dann
- wird Zugriff gegeben.



TCP-Wrapper

Konfigurationsdateien:
`/etc/hosts.allow`
`/etc/hosts.deny`

Reihenfolge der Auswertung:
Existiert ein passender Eintrag in der Datei /etc/hosts.allow, so wird Zugriff gegeben. Wenn nicht, dann Existiert ein passender Eintrag in der Datei /etc/hosts.deny, so wird kein Zugriff gegeben. Wenn nicht, dann wird Zugriff gegeben. Die Einträge werden der Reihe nach gelesen und der erste passende wird benutzt.

Die klassische Form der TCP-Wrapper (man 5 hosts_access)
Syntax:
Serverliste : Clientliste [: Shellkommando &]

- Serverliste ist eine Liste von Servern (Programmnamen), oder Wildcards. Server werden mit ihrem Programmnamen - nicht über ihr Protokoll - angegeben, also z.B.: **in.telnetd**
- Clientliste ist eine Liste von einem oder mehreren Hostnamen, IP-Adressen, Suchmustern oder Wildcards,
- Shellkommando ist ein Kommando, das die lokale Shell ausführt, wenn die Zeile zutrifft. Damit kann etwa eine Warnmeldung an root gegeben werden, wenn jemand versucht auf einen verbotenen Service zuzugreifen. Meist wird es zur Protokollierung eingesetzt. Es sollte grundsätzlich mit einem & beendet werden, weil sonst auf seine vollständige Abarbeitung gewartet wird, bevor ein Service evt. gestartet wird.

Suchmuster:

- Beginnt ein Suchmuster mit einem Punkt (z.B. .foo.bar), so gelten alle Hostnamen als Treffer, deren Ende mit dem Muster übereinstimmt also etwa hal.foo.bar
- Endet ein Suchmuster mit einem Punkt (z.B. 192.168.200.), so gelten alle Namen und Adressen als Treffer, deren erster Teil mit dem Muster übereinstimmt.

Wildcards:

ALL	Die universelle Wildcard, alles gilt...
LOCAL	Alle Hostnamen ohne Punkt (also lokale Namen) gelten.
UNKNOWN	Passt auf alle Usernamen, die unbekannt sind und alle Hosts, deren Namen oder Adressen nicht bekannt sind. Wird gerne in /etc/hosts.deny verwendet.
KNOWN	Passt auf alle Hosts und User, die bekannt sind
PARANOID	widersprüchliche Namensauflösung (z. B. Rechner mit mehreren Netzkarten)
EXEPT	Ist ein Operator, um zwei Listen auszuschließen also etwa ALL EXEPT UNKNOWN

zusätzliche Platzhalter in Shellkommandos:

%a	Die IP-Adresse des anfordernden Hosts
%A	Die IP-Adresse des aufgerufenen Servers
%c	Clientinformationen - User@Host oder User@IP-Adresse oder nur IP-Adresse des Anrufers, je nach dem, wieviel Informationen zur Verfügung stehen.
%d	Der Name des Daemon-Prozesses, der angefordert wurde.
%h	Name (oder falls nicht vorhanden IP-Adresse) des Clients
%H	Name (oder falls nicht vorhanden IP-Adresse) des Servers
%p	Die ProzessID des Daemon-Prozesses
%s	Serverinformationen - Daemon@Hostname oder Daemon@Adresse oder nur Daemon, je nach dem, wieviel Informationen zur Verfügung stehen.
%u	Der Username des Anrufers oder "unknown"
%%	Das %-Zeichen

Die modernere Form der Wrapper `(man 5 hosts_options)`
Syntax:
Serverliste : Clientliste [: Option] [: Option ...]
Optionen

ALLOW	Erlaubt den angegebenen Dienst für die angegebenen Clients.
DENY	Verbietet den angegebenen Dienst für die angegebenen Clients.

spawn *Shellkommando* Führt das angegebene Shellkommando aus. Platzhalter wie oben.
twist *Shellkommando* Führt das angegebene Shellkommando aus und schickt seine Ausgaben an den Client, anstatt den gewünschten Dienst zu starten. Platzhalter wie oben.
user *Username*[.*Gruppe*] Startet den angegebenen Dienst unter der angegebenen User (optional Gruppen) ID.
Vorteil:
Alle Einstellungen können in einer Datei vorgenommen werden. Mit ALLOW und DENY können in der Datei `/etc/hosts.allow` auch Verbote ausgesprochen werden und umgekehrt.

Bsp.:	in.ftpd : marvin.foo.bar : ALLOW	marvin darf
	in.ftpd : ALL : DENY	alle anderen nicht

Beispiele:

```
Example 1: Fire up a mail to the admin if a connection to the printer daemon
# has been made from host foo.bar.com, but simply deny all others:
# lpd : foo.bar.com : spawn /bin/echo "%h printer access" | \
#                               mail -s "tcp_wrappers on %H" root
#
#


# Example 2: grant access from local net, reject with message from elsewhere.
# in.telnetd : ALL EXCEPT LOCAL : ALLOW
# in.telnetd : ALL : \
#    twist /bin/echo -e "\n\raccess from %h declined.\n\rGo away.";sleep 2


#  Example 3: run a different instance of rsyncd if the connection comes
#            from network 172.20.0.0/24, but regular for others:
# rsyncd : 172.20.0.0/255.255.255.0 : twist /usr/local/sbin/my_rsyncd-script
# rsyncd : ALL : ALLOW

sshd : ALL EXCEPT 172.16.0.244: deny
```

/etc/hosts.allow

```
# Mail ist jedem gestattet
#
in.smtpd: ALL

# Telnet und FTP wird nur Hosts derselben Domain und
# dem Rechner "melmac" erlaubt.
#
in.telnetd, in.ftpd: LOCAL, melmac.outside.all

# Finger ist jedem erlaubt, aber root wird per Mail darüber informiert
#
ALL: ALL: spawn (/usr/sbin/safe_finger -l @%h | mail -s "finger from %h" root)
```

```
user@sonne> cat /etc/hosts.deny
ALL: ALL
```