Access Control Lists

1 Allgemeine Angaben in Regeln

1.1 Protokolle

ip – alle Protokolle
tcp – quittierte Verbindungen (www, telnet,
udp – nicht-quittierte Verbindungen (DNS, Streaming)
icmp – Ping usw
smtp – Mail-Server

1.2 Ports

ftp (21), www (80), dns (53), ssh (23), telnet
Rückkanal-Ports >1023 (z.B. für FTP, telnet)

1.2.1 Operatoren:

eq = gleich
gt = größer als
lt = kleiner als
neq = ungleich

1.3 Quell-/Zielangaben

Jeder Host	any
Spezieller Host	host [ip] [ip] 0.0.0.0
Rechnergruppe	[ip] [invertierte Netzmaske] Werte-mäßig kleinste IP wählen Maske erschließt sich aus variablen Bits
Subnetz	Im Grunde wie Rechnergruppe [netz-ip] [0.0.0.0 – 0.255.255.255]

1.4 Verbindungen allgemein

Anfrage auf well-known-port, Antwort meist auf Ports größer 1023

2 Standard ACLs

2.1 Anlegen

```
interface ethernet [eth-nr]
ip access-group [nr=0..99] [in|out]   ! beliebig oft
```

2.2 Regeln

access-list [nr] [permit|deny] [quelle]

2.3 ACL abschließen

```
deny any   # ist implizit
permit any
```

2.4 Beispiele

```
access-list 1 deny 172.16.4.13 0.0.0.0
access-list 1 permit 0.0.0.0 255.255.255.255
(implicit deny any)
(access-list 1 deny 0.0.0.0 255.255.255.255)

interface ethernet 0
ip access-group 1 out
```

3 Extended ACLs

3.1 Anlegen

```
interface ethernet [eth-nr]
ip access-group [nr=100..199] [in|out]     ! beliebig oft
```

Alternativ: ACL-Nummer, Interface und Richtung nennen

3.2 Regeln

```
access-list [nr] [permit|deny] [protokoll]
                 [quelle] {[op quellport]}
                 [ziel] {[op zielport]}
                 {log}
```

{} = optional

access-list [nr] kann ggf. auch weggelassen werden, wenn vorher ACL-Nummer, Interface und Richtung allgemein angegeben wurden

3.3 ACL abschließen mit

```
deny ip any any # alles verbieten, normal implizit, lieber angeben
permit ip any any # alles erlauben
```

3.4 Web-Server.

```
permit tcp [ziel] [webserver ip] eq www # Anfragen
permit tcp [webserver ip] eq www [ziel] gt 1023 # Antwort
```

3.5 Ping

Anfrage und Antwort beachten

```
permit icmp [quelle] [ziel] echo
permit icmp [quelle] [ziel] echo-reply
```

3.6 Beispiele

```
access-list 100 permit udp 192.168.1.0 0.0.0.255
                           host 192.168.2.2 eq domain
access-list 100 permit tcp 192.168.1.4 0.0.0.3
                           host 192.168.2.2 eq www
access-list 100 permit tcp host 192.168.1.4
                           host 192.168.2.3 eq telnet
access-list 100 permit tcp host 192.168.1.100
                           host 192.168.2.4 gt 1023

access-list 100 permit icmp any any echo
access-list 100 permit icmp any any echo reply
```

4 Named ACLs

4.1 Anlegen

```
ip access-list [standard|extended] [NAME]
[Regeln …]
interface ethernet [eth-nr]
ip access-group [NAME] [in|out]
```

4.2 Regeln

Wie bei Standard-ACLs bzw. Extended-ACLs, allerdings ohne „access-list [nr]“.

4.3 Beispiele

```
ip access-list extended Regel
permit TCP any 131.108.101.99 eq www
deny ip any any log
interface fastethernet 0/0
ip access-group Regel out
```

5 Reflexive ACLs

nur möglich für Extended Named ACLs.

5.1 Anlegen

Es werden zwei ACLs benötigt, eine für jede Richtung.

```
ip access-list [standard|extended] [ACL_NAME1]
[Regeln …]
ip access-list [standard|extended] [ACL_NAME2]
[Regeln …]
interface ethernet [eth-nr]
ip access-group [ACL_NAME1] in
ip access-group [ACL_NAME2] out
```

5.2 Regeln

Sie bestehen aus einem Regelpaar. Gedankenmodell: Verbindungsaufbau und –antwort. Die beiden zusammengehörenden Regeln stehen in verschiedenen ACLs und werden durch einen gemeinsamen Bezeichner(„REFNAME“) verbunden.

5.2.1 Aktion („Aufbau“)

```
[permit|deny] [protokoll]
[quelle] {[op quellport]}
[ziel] {[op zielport]}
reflect [REFMAME] {log}
```

{} = optional

5.2.2 Reaktion („Antwort)

```
evaluate [REFNAME]
```

5.3 Beispiel

```
ip access-list extended OUTBOUNDFILTERS
permit tcp 192.168.0.0 0.0.255.255 any reflect TCPTRAFFIC
permit icmp 192.168.0.0 0.0.255.255 any reflect ICMPTRAFFIC

ip access-list extended INBOUNDFILTERS
evaluate TCPTRAFFIC
evaluate ICMPTRAFFIC

interface S0/1/0
ip access-group INBOUNDFILTERS in
ip access-group OUTBOUNDFILTERS out
```