IT-Sicherheit Praktikum Nr.4:

Firewall: Accesslisten auf Cisco Routern

Eine Firewall ist ein Gerät, das eine Filterfunktion zur Verfügung stellt, um ein Datennetz vor unerwünschten Zugriffen zu schützen. Es gibt verschiedene Arten von Firewalls, *stateless*, *statefull* uvm. Der größte Teil dieser Typen kann auch auf Cisco Routern, über ein spezielles IOS (Firewall IOS), eingesetzt werden. Welche Typen das sind und wie sie konfiguriert und eingesetzt werden, soll in diesem Praktikum gezeigt werden. Dabei werden ausschließlich Typen vorgestellt, die auf TCP/IP aufsetzen. Natürlich gibt es auch für andere Protokolle passende Filter!

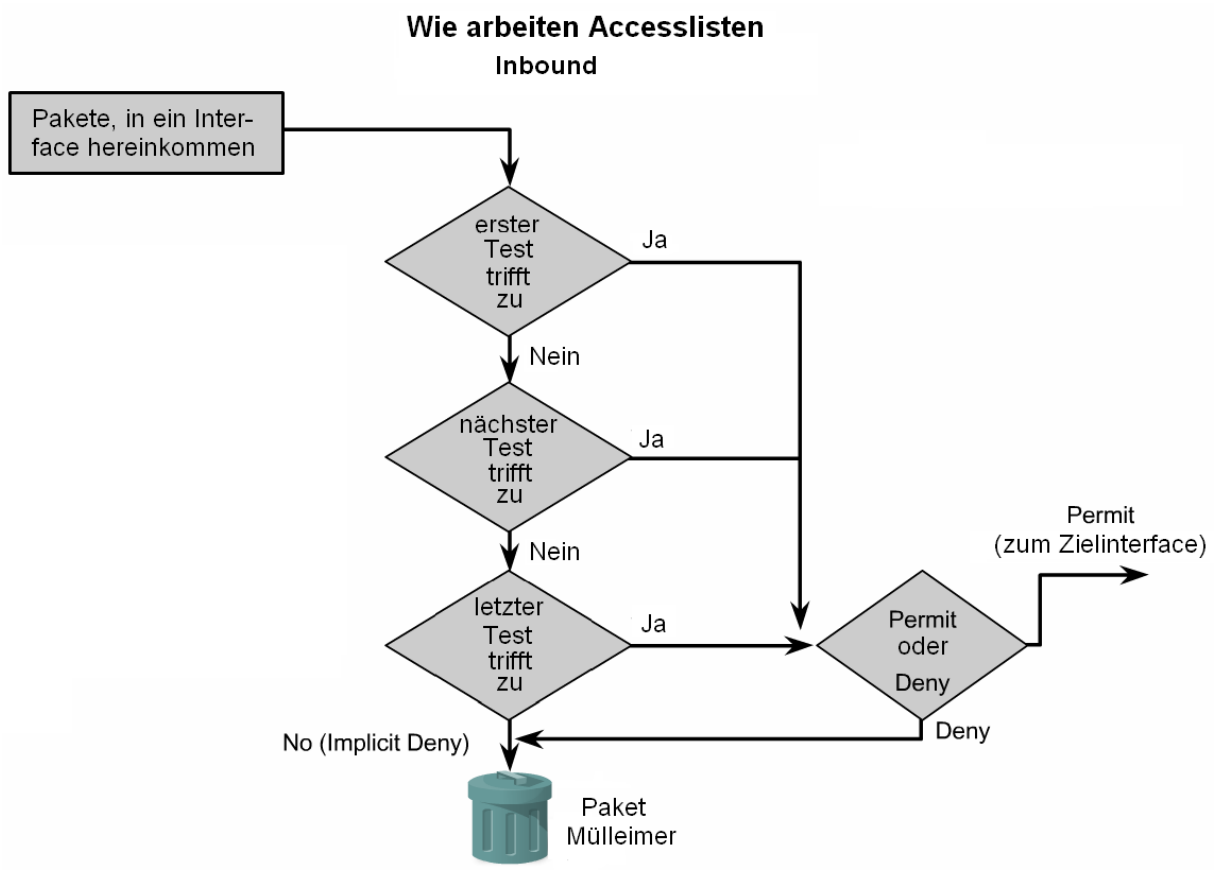
Die Filter werden in der Cisco-Welt Accesslisten genannt. Auf einem Cisco Router können u.a. folgende Accesslistentypen eingesetzt werden:

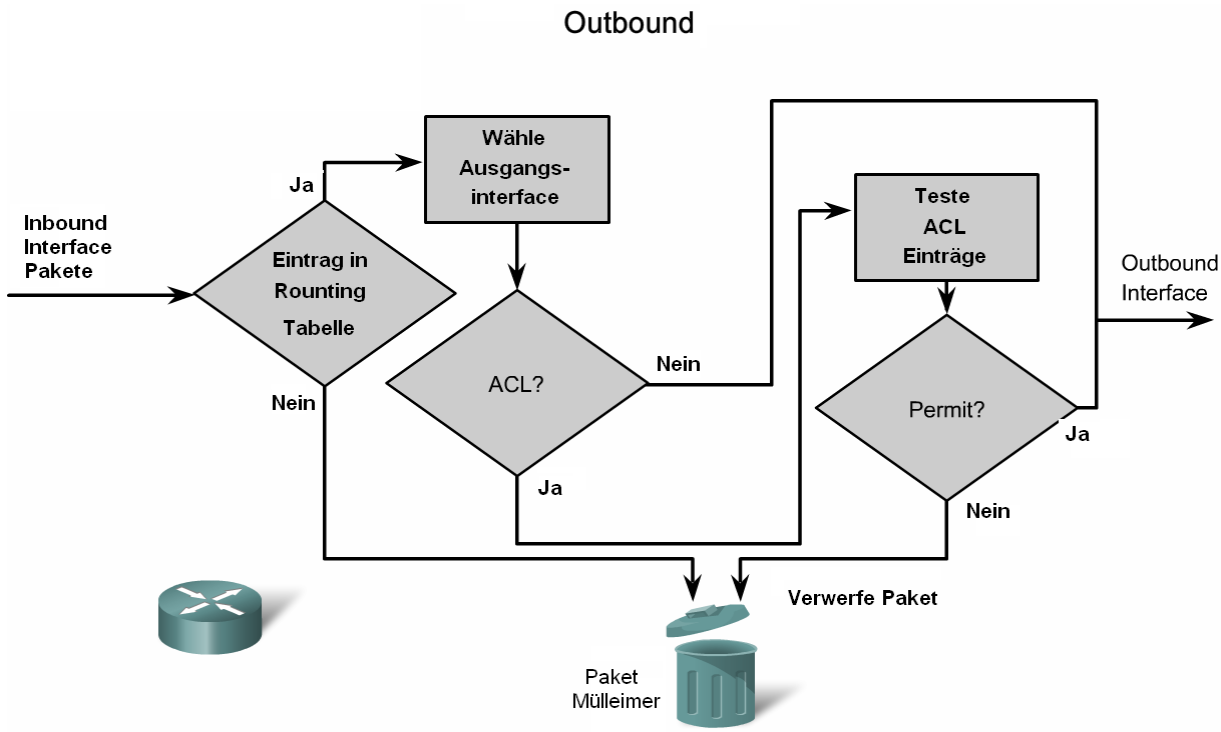
- Standard Accessliste (standard access-list)
- Erweiterte Accessliste (extended access-list)
    - Nummern Accessliste (numbered access-list)
    - Namens Accessliste (named access-list)
- Zeitbasierte Accessliste (time based access-list)
- Dynamische Accessliste (dynamic access-list)
- Reflexive Accessliste (reflexive access-list)

Wie diese Filter funktionieren, soll im Folgenden kurz beschrieben und von Ihnen ausprobiert werden.

Standard Accesslisten

Eine Standard Accessliste (oft auch Paket Filter genannt), setzt auf der 3. OSI Schicht, der Netzwerkschicht, auf. Sie arbeitet mit den IP Adressen im IP Header und kann einzelnen Rechnern, oder ganzen Subnetzen, den Zutritt in einen Bereich, bzw. das Verlassen eines Bereiches, erlauben oder verbieten. Dabei nennt man den Eingang in einen Bereich ***Inbound*** und das Verlassen eines Bereichs ***Outbound***. Accesslisten werden Schnittstellen zugeordnet und verarbeiten dann alle Pakete, die in diese Schnittstelle hinein gehen (Inbound) oder herauskommen (Outbound). Wie der Name schon sagt, handelt es sich um Listen, d.h. es werden unterschiedliche Filterfunktionen zu einer Liste zusammen gefasst und dann wie ein Filter betrachtet. Die einzelnen Listeneinträge werden vom Router nacheinander abgearbeitet. Sobald ein Listeneintrag auf ein bearbeitetes Paket zutrifft, wird das Paket entsprechend behandelt, z.B. verworfen (deny) oder durchgelassen (permit). Die folgenden Listeneinträge werden dann nicht mehr auf das Paket angewendet. Am Ende einer jeden Accessliste steht IMMER (nicht sichtbar) ein impliziter, nicht löschbarer Eintrag, der alle Pakete verwirft (also nicht passieren lässt). Eine Standard Accessliste öffnet ein permanentes Loch in der Firewall.



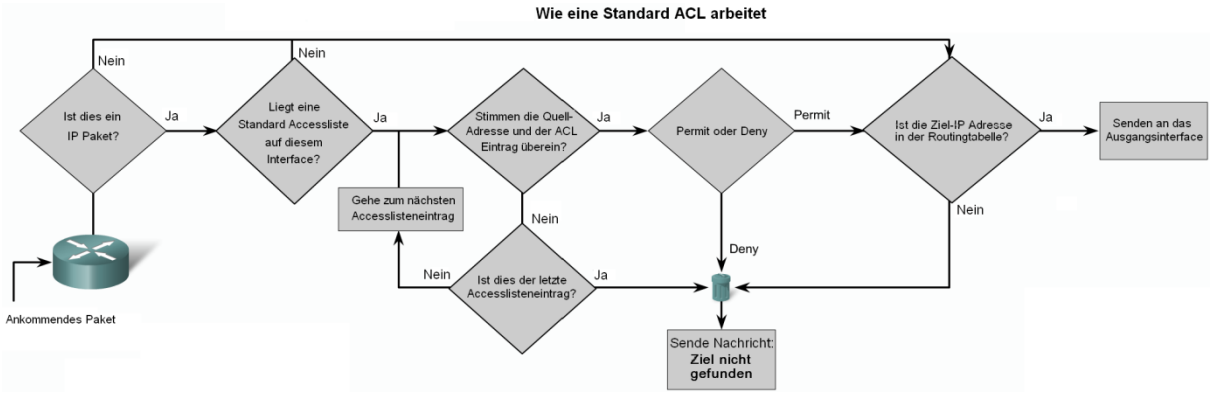


Eine Standard Accessliste filtert nur basierend auf der Quell-IP Adresse. Die Zieladresse und die TCP Ports spielen bei der Filterung keine Rolle.

Der Cisco Befehl sieht folgendermaßen aus:

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

Der Befehl **access-list** kennzeichnet einen Accesslisteneintrag.
Die **10** identifiziert eine Accessliste. Alle Accesslisteneinträge mit der Nummer 10 gehören zu einer Accessliste. Für Standard Accesslisten sind die Nummern 1-99 und 1300-1999 vorgesehen.
permit erlaubt Paketen, die durch die folgende IP Adresse identifiziert werden, den Filter zu passieren. An dieser Stelle kann auch **deny** stehen, dann würde das Paket in den Paket Mülleimer geworfen.
192.168.30.0 0.0.0.255 kennzeichnet die IP Adressen, auf die dieser Accesslisteneintrag zutrifft. Dabei ist der 2. Teil (0.0.0.255) eine Wildcard Maske. Bei einer 0 in der (binären) Wildcard Maske muss die zu vergleichende IP Adresse (die sich als Quelladresse im untersuchten IP Paket befindet) in diesem Bit genau mit dem im Accesslisteneintrag benannten IP Adressenbit übereinstimmen. Bei einer 1 in der Wildcard Maske ist der Wert in der IP Adresse unerheblich. Im oben gezeigten Beispiel werden also alle Pakete mit den Quell-IP Adressen 192.168.30.0 – 192.168.30.255 erlaubt.



Um einem Interface eine Accessliste zuzuweisen wird der folgende Befehl verwendet.

```
R1(config)# interface FastEthernet 0/0
R1(config-if)# ip access-group 1 out
```

Zuerst wird das Interface gewählt. Dann wird die Accessliste zugewiesen. Die Zahl 1 ist die Nummer der Accessliste. **out** bedeutet Outbound, **in** entsprechend Inbound.

Jetzt noch einmal der komplette Konfigurationsablauf für eine Standard Accessliste.

Schritt 1:
Mit dem **access-list** Befehl wird im globalen Konfigurationsmodus ein Accesslisteneintrag für eine IPv4 Accessliste erzeugt.

```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

Um eine Accessliste zu löschen verwenden Sie den **no access-list** Befehl.
Mit der Option **remark** kann eine Beschreibung zur Accessliste zugefügt werden.

Schritt 2:

Mit dem interface Befehl wird das Interface ausgewählt, dem die Accessliste zugeordnet werden soll.

```
R1(config)# interface FastEthernet 0/0
```

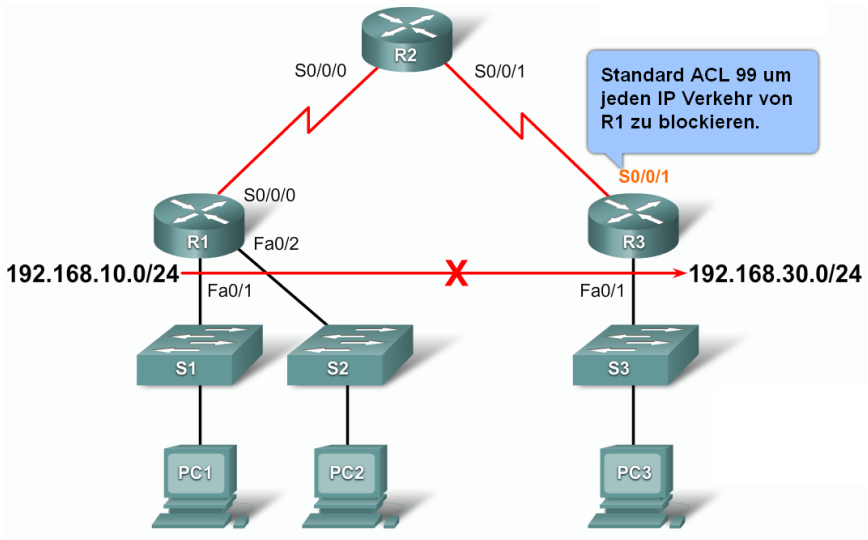
Schritt 3:

Mit dem Befehl **ip access-group** im Interface Konfigurationsmodus wird die Accessliste auf dem Interface aktiviert.

```
R1(config-if)# ip access-group 1 out
```

Um eine Accessliste von einem Interface zu entfernen, verwenden Sie den Befehl **no ip access-group**.

Da eine Standard Accessliste nur über die Quell-IP Adresse filtert, muss sie so nah wie möglich am Ziel konfiguriert werden, da sonst eventuell auch andere Ziele abgeschnitten würden. Im folgenden Bild wird das weiter verdeutlicht.



Erweiterte Accesslisten

Erweiterte Accesslisten arbeiten auf der 4. Schicht des OSI Referenzmodells. Sie verarbeiten Quell- und Ziel-IP Adressen, Quell- und Zielport, sowie unterschiedliche Protokolle (z.B. icmp, TCP und UDP). Wie die Standard Accessliste öffnet sie ein permanentes Loch in der Firewall.
Im Bild unten wird der Konfigurationsbefehl für eine erweiterte Accessliste gezeigt.

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

Der Befehl access-list hat die gleiche Wirkung wie bei der Standard Accessliste. Die Zahl 103 identifiziert wieder die Accessliste (für erweiterte Accesslisten sind die Zahlen 100-199 und 2000 – 2699 vorgesehen). Für erweiterte Accesslisten stehen die Zahlen 100-999 zur Verfügung. **permit**, bzw **deny** legen wieder die Filterfunktion fest. Darauf folgt das zu filternde Protokoll **tcp**. Hier kann auch **ip**, **udp** oder **icmp** stehen. Es kann aber auch eine Protokollnummer, z.B. für Routingprotokolle usw.) angegeben werden. Bei der Standard Accessliste musste kein Protokoll angegeben werden, da es nur auf IP arbeitet.
Nach dem Protokoll folgt die Quelladresse. Sie wird genauso, wie bei der Standard Accessliste angegeben (IP Adresse + Wildcard Maske). Darauf folgt die Zieladresse. Hier im Beispiel soll jede beliebige Adresse gültig sein, dazu gibt es das Kennwort **any**. **any** kann auch für Quell-IP Adressen eingesetzt werden. Soll nur ein einzelner Rechner angegeben werden gibt es 2 Möglichkeiten dies auszudrücken.

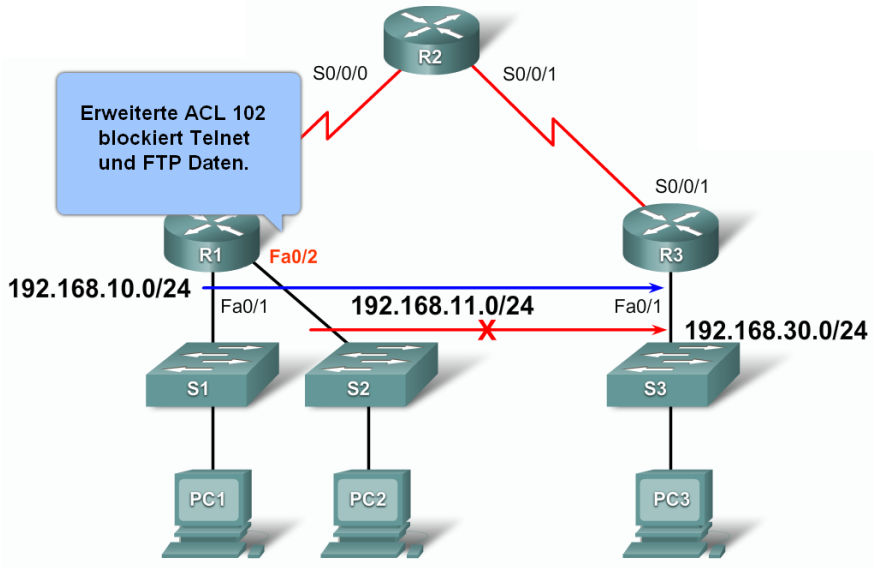
1. 192.168.30.1 0.0.0.0 oder
2. host 192.168.30.1

Unter 1. besteht die Wildcard Maske nur aus Nullen, d.h. die zu untersuchende IP Adresse muss komplett mit der in der Accessliste angegebenen IP Adresse übereinstimmen. Unter 2. Wird die WIldcardmaske weg gelassen und das Kennwort **host** vor die IP Adresse gestellt. Beide Schreibweisen sind gleichwertig einsetzbar.
Am Ende des Konfigurationsbefehls wird noch ein Port angegeben. Im Beispiel oben muss der Zielport gleich 80 (equal) sein. Es kann aber nicht nur auf Gleichheit abgefragt werden. Es gibt auch die Parameter **lt** (lower than, kleiner als), **le** (lower equal, kleiner gleich), **gt** (greater than, größer als), **ge** (greater equal, größer gleich) und **ne** (not equal, ungleich). So können also auch ganze Portgruppen verarbeitet werden. Soll ein Quellport angegeben werden, so muss dieser hinter der Quell-IP Adresse angegeben werden, z.B.

access-list 100 permit tcp host 100.100.100.1 eq 80 any

Hier wird einem Webserver mit der IP Adresse 100.100.100.1 an alle Rechner seine HTTP-Pakete zu senden.

Im Gegensatz zu Standard Accesslisten werden erweiterte Accesslisten möglichst nah am Ziel konfiguriert. Da sowohl Quell- als auch Zieladresse verarbeitet werden, können gezielt Pakete gefiltert werden.



Für alle Accesslisten gilt noch folgende Regel:

Pro Protokoll (IP, TCP, UDP…) kann, pro Interface und Richtung (Inbound/Outbound), nur eine Accessliste zugewiesen werden!

Da die Verwendung der Wildcard Masken nicht trivial ist, sollen jetzt einige Beispiele für die Ermittlung einer bestimmten Maske folgen.

Beispiel1:

Sie verfügen über folgende IP Adressen: 149.201.41.0/255 und wollen den IP Adressen 149.201.41.32-149.201.41.63 verbieten ins Internet zu gehen. Sie erzeugen eine entsprechende erweiterte Accessliste:

access-list 100 deny ip IP_ADR WCM any

Welche Werte müssen für IP_ADR und WCM eingesetzt werden?

Dieser Wert beinhaltet immer die Startadresse der zu filternden Geräte, in diesem Fall 149.201.41.32. Nun sollen insgesamt 32 Geräte gefiltert werden. Es gibt einen einfachen Trick, die Wildcard Maske zu ermitteln. Da wir uns viel besser mit Subnetzmasken auskennen, verwenden wir diese zur Berechnung der Wildcard Maske. Eine Subnetzmaske, die 32 Geräte maskiert sieht (im letzten Oktett binär so aus: 00011111. Die gesamte Subnetzmaske dezimal sieht demnach folgendermaßen aus: 255.255.255.224. Zur Berechnung der Wildcard Maske wird folgende Berechnung durchgeführt.

```
      255.255.255.255
  -   255.255.255.224
-----------------------------------
       0 . 0 . 0 . 31
```

Somit lautet der gesamte Befehl nun:

access-list 100 deny ip 149.201.41.32 0.0.0.31 any

Auf diese Weise können Bereiche, die auf 2er-Potenzen starten und eine 2er-Potenz-Anzahl von IP Adressen beinhaltet, definiert werden. Andere Bereiche müssen dementsprechend unterteilt werden.

Noch ein weiteres Beispiel (ehre akademischer Art, da das in der realen Welt nicht vorkommt):

Sie haben den oben genannten IP Adressbereich und sollen alle geraden IP Adressen dieses Bereiches ausfiltern. Da es hier keinen zusammenhängenden IP Bereich gibt, kann auch das oben gezeigte Verfahren nicht angewendet werden.

Zur Lösung untersuchen wir die zu filternden Adressen. Eine gerade (binäre) Zahl zeichnet sich dadurch aus, dass das 1er Bit immer 0 ist.

Der Bereich ist definiert durch 149.201.41.0 0.0.0.255
Das 1 Bit des letzten Oktetts muss aber 0 sein. Das ist bei der IP Adresse im Accesslisteneintrag schon der Fall, allerdings erlaubt die Wildcard Maske durch eine 1, dass dieses Bit nicht mit der IP Adresse übereinstimmen muss. Ändern wir die Wildcard Maske auf 0.0.255.254, so muss das 1er Bit immer Null sein. Der Accesslisteneintrag sieht dann folgendermaßen aus:

access-list 100 deny ip 149.201.41.0 0.0.0.254 any

Denken Sie einmal darüber nach …

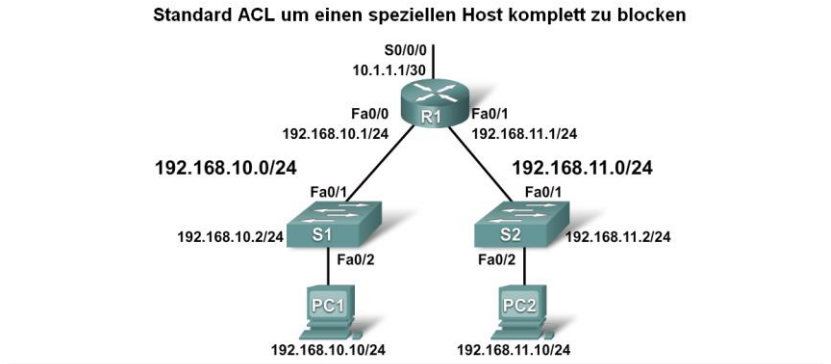
Wichtiger Hinweis:

Alle Accesslisten haben, wie oben schon einmal erwähnt, am Ende ein implizites **deny any**. Das heißt, dass alle Pakete, auf die bisher kein Accesslisteneintrag zugetroffen ist, verworfen werden.
Weiterhin bedeutet das, dass eine Accessliste, die nur deny (oder auch gar keinen einzigen) Accesslisteneinträge hat, alle Pakete verwirft, also die Schnittstelle komplett blockiert.

Zur Übung sollen noch einmal 2 Beispiele für die Konfiguration von Accesslisten gezeigt werden.

1. Beispiel: Standard Accessliste

Im unten gezeigten Netz soll PC1 nicht ins Internet (hängt an S0/0/0) kommen. Der zweite Accesslisteneintrag ist notwendig, da sonst auf Grund des impliziten **deny any** die gesamte Schnittstelle blockiert wäre!

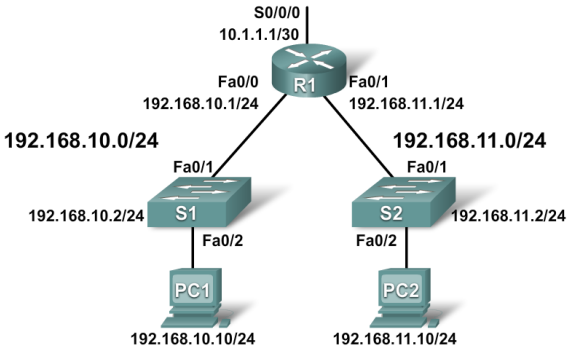



```
R1(config)#no access-list 1
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 1 out
```

2. Beispiel: Erweiterte Accessliste

Im unten gezeigten Netz soll PC1 nicht auf Webserver im Internet (über S0/0/0) kommen.

Erweiterte ACL um den Webzugriff eines speziellen Hosts zu blocken




```
R1(config)#no access-list 100
R1(config)#access-list 100 deny host 192.168.10.10 any eq 80
R1(config)#access-list 100 permit 192.168.10.0 0.0.0.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 100 out
```

Konfigurierte Accesslisten können Sie sich mit dem folgenden Befehl ansehen:

```
R1# show access-lists {access-list-number}
```

Alle bisher gezeigten Accesslisten wurden durch Nummern identifiziert. Weiterhin wurden die Accesslisteneinträge in der Reihenfolge der Eingabe abgespeichert. Damit ist eine Korrektur nur möglich, indem die gesamte Liste gelöscht und dann mit der Korrektur erneut eingegeben wird. Da ist etwas mühselig. Weiterhin ist die Aussagekraft einer Zahl sehr gering (siehe IP-Adresse/Domainname). Aus diesen Gründen wurden die Namensaccesslisten eingeführt.

Namensaccesslisten

Beispiel einer Namensaccessliste

```
Router(config)# ip access-list [standard | extended] name
```

- Der Alphanumerische Name muss eindeutig sein un darf nicht mit einer Zahl beginnen

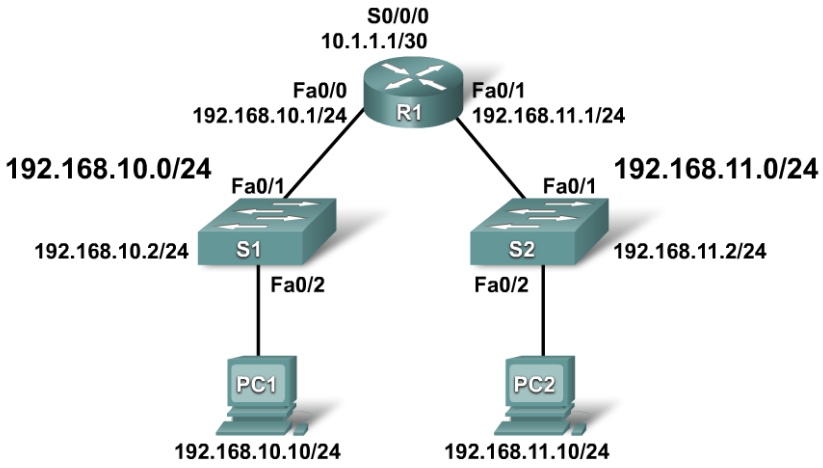
```
Router(config-std-nacl)# [permit | deny | remark] {source [source-
wildcard]} [log]
```

- Falls nicht konfiguriert werden automatisch Sequenznummern erzeugt (starten mit 10 und erhöhen immer um 10
- Mit dem Befehl **no** *Sequenznummer* wird der entsprechende Accesslisteneintrag aus der ACL gelöscht

```
Router(config-if)#ip access-group name [in | out]
```

- Aktiviert die Namensaccessliste auf einem Interface

Im Beispiel oben wird eine Standard Namensaccessliste erzeugt (beachten Sie in welchem Modus Sie sich in der 2. Zeile befinden!).

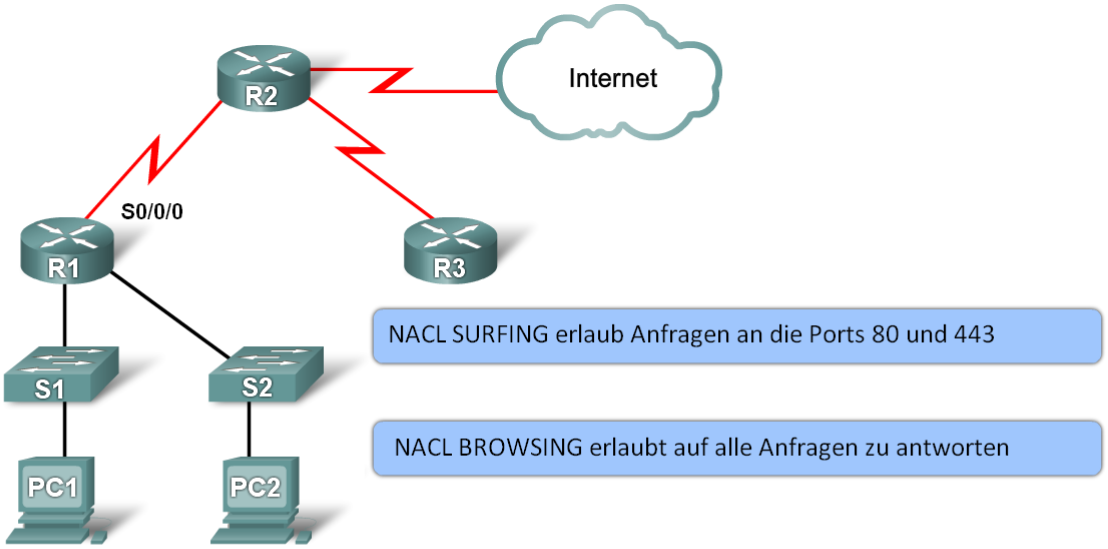



```
R1(config)#ip access-list standard NO_ACCESS
R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#permit 192.168.11.0 0.0.0.255
R1(config-std-nacl)#interface Fa0/0
R1(config-if)#ip access-group NO_ACCESS out
```

Um einzelnen Accesslisteneinträgen Nummern zu geben und sie zwischen schon vorhandenen Einträgen zu platzieren, konfigurieren Sie folgendermaßen:

```
R1# show access-lists
Standard IP access list WEBSERVER
    10 permit 192.168.10.10
    20 deny   192.168.10.0, wildcard bits 0.0.0.255
    30 deny   192.168.11.0, wildcard bits 0.0.0.255
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip access-list standard WEBSERVER
R1(config-std-nacl)# 15 permit host 192.168.11.10
R1(config-std-nacl)# end
R1#
*Nov  1 19:20:57.591: %SYS-5-CONFIG_I: Configured from console by console
R1# sho access-lists
Standard IP access list WEBSERVER
    10 permit 192.168.10.10
    15 permit 192.168.11.10
    20 deny   192.168.10.0, wildcard bits 0.0.0.255
    30 deny   192.168.11.0, wildcard bits 0.0.0.255
R1#
```

Erweiterte Namensaccesslisten werden folgendermaßen konfiguriert:




```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.225 any eq 443
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
```

Der Parameter *established* bezieht sich auf die SYN und ACK Flags von TCP. Bei der ersten Anfrage des TCP Verbindungsaufbaus ist NUR das SYN Flag gesetzt. Jede weitere Antwort hat das ACK Flag gesetzt. Mit dem Parameter *established* werden alle TCP Verbindungen, die NICHT das ACK Flag gesetzt haben, ausgefiltert. Diese Methode ist sehr einfach (und auch einfach auszutricksen), wir werden später bessere Verfahren kennenlernen.

Zeitbasierte Accesslisten

Zeitbasierte Accesslisten aktivieren konfigurierte Accesslisten zu einer bestimmten Zeit mit einer bestimmten Dauer. Die Konfiguration von zeitbasierten Accesslisten geschied folgendermaßen:

Schritt 1: Zuerst werden der Zeitraum, an dem die Accessliste aktiv sein soll und ihr Name festgelegt. Im Beispiel unten wird eine periodische Zeit angegeben (jeden Montag…). Es können auch Einzeltermine konfiguriert werden.

```
R1(config)#time-range EVERYOTHERDAY
R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to
17:00
```

Schritt 2: Der definierte Zeitraum wird über den Namen einer Accessliste hinzugefügt, indem Sie einfach hinten angehängt wird.

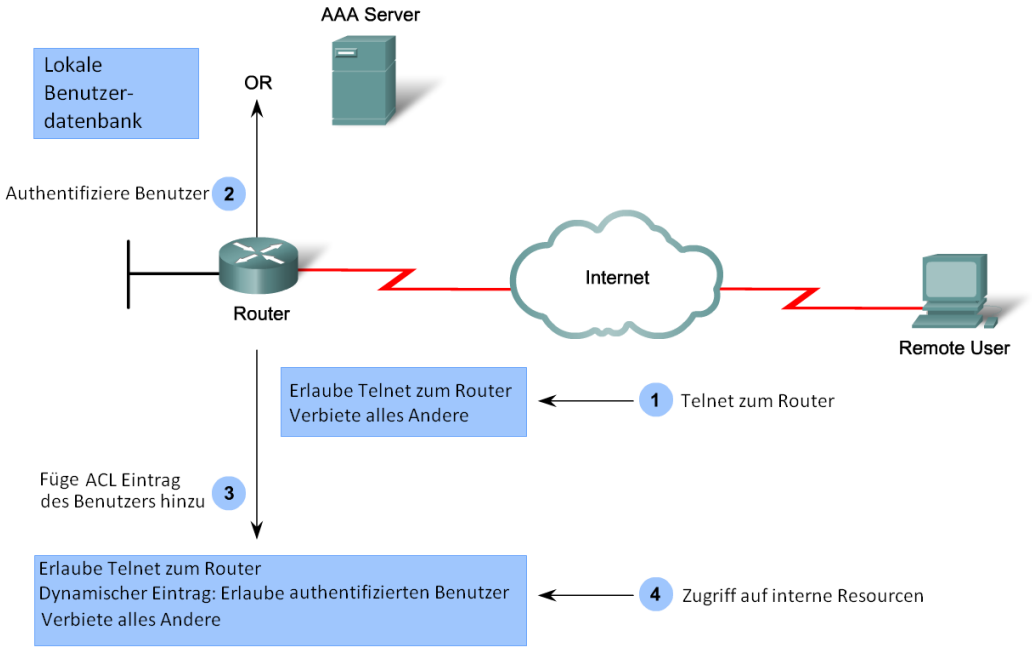
```
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255
any eq telnet time-range EVERYOTHERDAY
```

Schritt 3: Die Accessliste wird einem Interface zugeordnet.

```
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 101 out
```

Dynamische Accesslisten

Dynamische Accesslisten (auch Lock and Key genannt) sind Accesslisten, die erst nach einer Authentifizierung aktiv werden. Dazu muss sich der entsprechende Benutzer mit dem Router per Telnet (oder SSH) verbinden und sich authentifizieren. Direkt danach wird die dynamische Accessliste aktiviert. So können permanente Löcher in der Firewall vermieden werden.



Die Konfiguration einer dynamischen Accessliste geschied folgendermaßen:

Schritt 1: Erzeugen eines Benutzers, der die dynamische Accessliste aktivieren kann.

```
R3(config)#username Student password 0 cisco
```

Schritt 2: Erweiterte Accessliste, die den Zugriff über Telnet auf den Router zu lässt. Die dynamische ACL ist deaktiviert bis zur Authentifizierung des Benutzers.

```
R3(config)# access-list 101 permit tcp any host 10.2.2.2
eq telnet
R3(config)#access-list 101 dynamic testlist timeout 15
permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
```

Schritt 3: Zuweisen der Accessliste zu einem Interface.

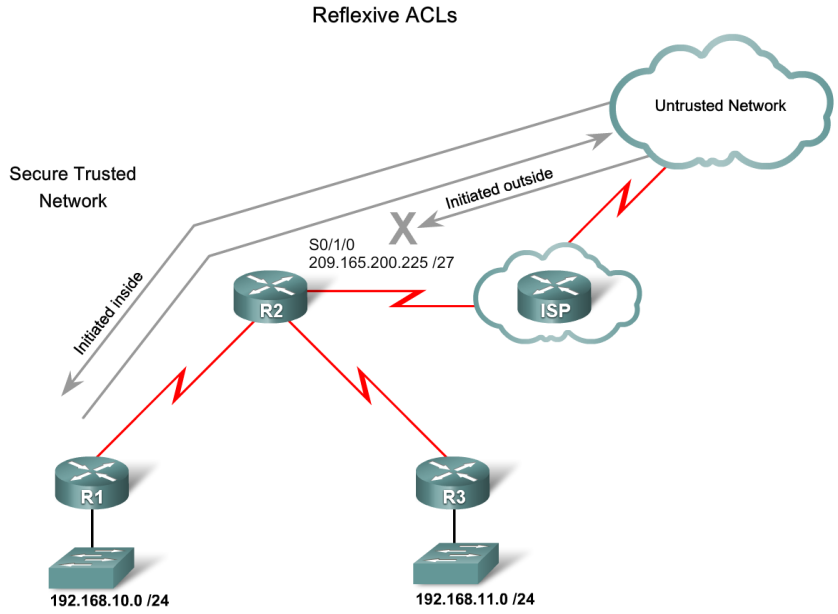
```
R3(config)#interface serial 0/0/1
R3(config-if)#ip access-group 101 in
```

Schritt 4: Nachdem der Benutzer sich mit Telnet authentifiziert hat, wird der autocommand Befehl ausgeführt und die Telnetsession geschlossen. Der Benutzer kann nun auf das Netz zugreifen. Nach 5 Minuten ohne Aktivität wird die Accessliste deaktiviert.

```
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)# autocommand access-enable host timeout 5
```

Reflexive Accesslisten

Eine reflexive Accessliste erlaubt die Antwort vom Ziel auf ein von der Quelle gesendetes Paket (ähnlich wie der Parameter *established* bei einer erweiterten Accessliste). Allerdings wird hier nicht auf die TCP Flages geachtet, sondern der Start (fast immer ein Outbound Paket) einer Verbindung wird in eine Tabelle eingetragen (Quell IP, Ziel IP, Quell Port und Ziel Port) und das zugehörige Antwortpacket kann dann die Firewall passieren.



Durch diese Accessliste wird also kein permanentes Loch in die Firewall gebohrt, sondern ein interner Benutzer kann z.B. zu einen externen Webserver browsen und die Antwort kann zu ihm zurück kommen, währen der gleiche Webserver keine Verbindung zu diesem Benutzer starten kann.

Schritt 1: Veranlasst den Router Verbindungen zu beobachten, die vom Netzinneren gestartet werden.

```
R2(config)#ip access-list extended OUTBOUNDFILTERS
R2(config-ext-nacl)# permit tcp 192.168.0.0 0.0.255.255
any reflect TCPTRAFFIC
R2(config-ext-nacl)# permit icmp 192.168.0.0 0.0.255.255
any reflect ICMPTRAFFIC
```

Schritt 2: Erzeugt eine Inbound-Richtlinie, die den Router veranlasst, Daten, die zu einer von Innen initiierten Verbindung gehören, passieren zu lassen.

```
R2(config)#ip access-list extended INBOUNDFILTERS
R2(config-ext-nacl)# evaluate TCPTRAFFIC
R2(config-ext-nacl)# evaluate ICMPTRAFFIC
```

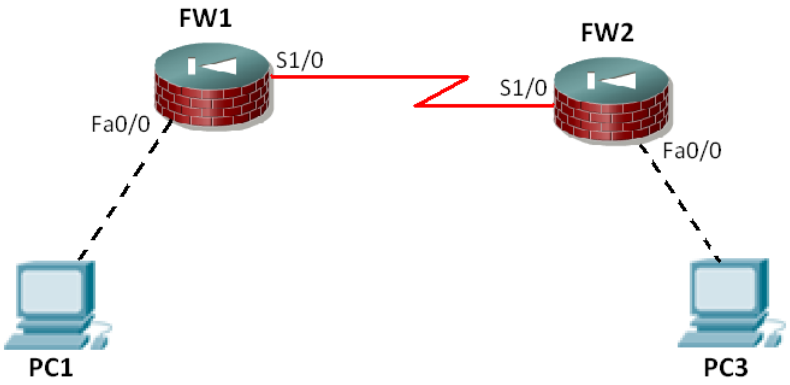
Schritt 3: Weist die (inbound und outbound) Accesslisten einem Interface zu.

```
R2(config)#interface S0/1/0
R2(config-if)#ip access-group INBOUNDFILTERS in
R2(config-if)#ip access-group OUTBOUNDFILTERS out
```

Eine reflexive Accessliste erzeugt nur, als Reaktion auf eine vom sicheren Netz gestartete Netzverbindung, ein temporäres Loch in die Firewall. Für einen Angreifer ist es viel schwieriger, eine Firewall mit reflexiven Accesslisten zu überwinden, als eine mit Standard-, oder erweiterten Accesslisten.

Testen Sie nun die oben vorgestellten Accesslisten.

Bauen Sie ein Netz gemäß Bild unten auf. Die mit **FW** bezeichneten Geräte sind Router mit Firewall Betriebssystemen. Konfigurieren Sie die Schnittstellen gemäß der Tabelle unten. Konfigurieren Sie auf jedem Router eine Defaultroute zum gegenüberliegenden Gerät (also von FW1 zu FW2 und umgekehrt).



Konfigurieren Sie die Geräte gemäß der Tabelle.

	S1/0	Fa0/0	NIC
FW 1	10.0.0.1/24	192.168.1.1/24	
FW2	10.0.0.2/24	192.168.2.1/24	
PC1			192.168.1.2/24
PC2			192.168.2.2/24

1. Nachdem Sie die Router und die PCs konfiguriert haben, pingen Sie von PC1 zu PC2. Falls der Ping nicht erfolgreich ist, führen Sie eine Fehlersuche durch.

2. Erzeugen Sie eine Nummern-Standard Accessliste, die die Verbindung von PC1 zu PC2 verbietet, sonst aber jegliche Daten passieren lässt.
   Wie lautet der Befehl?

   access-list 1 deny host 192.168.1.2
   access-list 1 permit any any

3. Welcher Schnittstelle von welchem Router in welche Richtung muss diese Accessliste zugewiesen werden?

   Interface Fa0/0 von FW2 outbound

4. Wie lautet der entsprechende IOS Befehl?

   Interface Fa0/0
   Ip access-group 1 out

5. Zeigen Sie die konfigurierte Accessliste an, wie lautet der Befehl?

   show access-list 1

6. Wenn Sie alles richtig konfiguriert haben, sehen Sie nun die Accessliste in der Ausgabe.

7. Führen Sie erneut ein Ping von PC1 zu PC2 durch, er sollte nicht funktionierten!

8. Wenn Sie nicht wüssten, dass eine Accessliste konfiguriert wäre (und bei großen und langen Konfigurationen kann das schon einmal passieren!), würden Sie nun eine Fehlersuche starten. Schöner wäre es, wenn Sie die Firewall unterrichtet, dass auf Grund einer Accessliste ein Paket gefiltert wurde. Dass können Sie erreichen. Löschen Sie die Accessliste 1 und geben Sie sie erneut ein, allerdings geben Sie am Ende jedes Accesslisteneintrages ein *log* ein (z.B. **access-list 1 permit any log**). Führen Sie nun erneut ein Ping von PC1 zu PC2 durch. Welche Meldung gibt die Firewall nun aus?

***Mar 1 00:21:07.023: %SEC-6-IPACCESSLOGNP: list 1 denied 0 192.168.1.2 -> 192.168.2.2, 4 packets**

9. Zeigen Sie nun auf FW2 erneut die Accesslist 1 an. Was hat sich geändert?

   Die Anzahl der durch einen Accesslisteneintrag gefilterten Pakete wird angezeigt.

10. Die Standard Accessliste lässt keine Daten von PC2 in das Netz 192.168.2.0/24 durch, unabhängig vom Ziel (es wird ja nur nach der Quell-IP-Adresse gefiltert. Präziser kann das eine erweiterte Accessliste. Löschen Sie die Standard Accessliste (auch vom Interface Fa0/0) und ersetzten Sie sie durch eine erweiterte Accessliste. Prüfen Sie nach dem Löschen, dass PC1 PC3 wieder pingen kann. Wo wird diese Accessliste konfiguriert, welchen Befehl verwenden Sie und an welche Schnittstelle in welche Richtung wird sie gebunden. Geben Sie den entsprechenden Befehl an (verwenden Sie wieder den *log* Parameter am Ende!)

    access-list 100 deny ip host 192.168.1.2 host 192.168.2.2 log
    access-list 100 permit ip any any log

    int fa0/0
    ip access-list 100 in

11. Pingen Sie das FastEthernet Interface von FW2. Diese IP Adresse sollte erreichbar sein (Sie haben ja nur das Ziel PC2 gefiltert!). Sollte es nicht klappen, führen Sie eine Fehlersuche durch!

12. Jetzt soll die gleiche ACL als Namens ACL mit dem Namen INFOSEC konfiguriert werden. Löschen Sie die ACL und die Zuordnung zum Interface und konfigurieren Sie sie neu. Wie sieht die Konfiguration aus?

```
ip access-list extended INFOSEC
 deny ip host 192.168.1.2 host 192.168.2.2 log
 permit ip any any log
int fa0/0
 ip access-group INFOSEC
```

13. Testen Sie die Accessliste. Pingen Sie einmal PC2 von PC1 aus, dann einmal mit FW1 als Quelle.

14. Nun soll diese Accessliste in eine zeitbasierte Accessliste geändert werden. Löschen Sie dazu zuerst die beiden Einträge (deny und permit) in der Liste.

15. Wechseln Sie nun in den allgemeinen Konfigurationsmodus und erzeugen Sie eine Zeitreference mit dem Befehl **time-range** mit dem Namen **INFOSECTIME**. Konfigurieren Sie ein Zeitfenster von 10:00 Uhr bis 10:01 Uhr am heutigen Datum. Sie können sich mit dem **?** durch die Befehlsparameter hangeln. Lesen Sie sich die möglichen Parameter und deren Funktion durch. Wie lauten die benötigten Befehle?

    time-range INFOSECTIME
    absolute start 10:01 30 July 2010 end 10:02 30 July 2010

16. Nun soll die Names Accessliste so konfiguriert werden, das kein Zugriff vom 192.168.1.0/24 Netz auf den Host 192.168.2.2 möglich ist. Nur zur Zeit 10.01 – 10.02 Uhr am heutigen Datum soll der Host 192.168.1.2 Zugriff auf diesen Host bekommen.

    Konfigurieren Sie in der Accessliste **INFOSEC** den benötigten Eintrag, wie lautet er? (fügen Sie wieder den log-Parameter an!)

    ip access-list extended INFOSEC
    permit ip host 192.168.1.2 host 192.168.2.2 time-range INFOSECTIME log

17. Um die Funktion der Accessliste zu testen stellen wir nun die Routerzeit auf 10.00 Uhr des heutigen Datums ein. Beim folgenden Beispiel wird als Datum der 6.12.2010 verwendet, passen Sie ihn entsprechend an! Geben Sie folgenden Befehl im privilegierten Modus ein:

    FW1# clock set 10:00:00 6 December 2010

    Prüfen Sie mit **show clock** die eingestellte Zeit,

18. Um die Accessliste zu testen öffnen Sie auf PC1 eine DOS-Box und geben Sie einen dauerhaften ping Befehl auf PC2 ein.

    C:\> ping 192.168.2.2 –t

    Bevor der Ping erfolgreich ist, schauen Sie sich die Accessliste mit dem Befehl show access-lists an. Was steht bei der zeitbasierten Accessliste? **inactive**

    Prüfen Sie die Zeit auf dem Router, ab der der Ping erfolgreich ist. Zeit: _____

    Wenn der Ping erfolgreich ist, was steht nun bei der zeitbasierten Accessliste? **active**

    Prüfen Sie die Zeit auf dem Router, ab der der Ping nicht mehr erfolgreich ist. Zeit: _____

19. Zwar erzeugt die zeitbasierte Accessliste nicht ein dauerhaftes Loch in der Firewall, ist aber dennoch ein Packet Filter, d.h. speichert keine Zustand der Verbindungen und das zeitlich

begrenzte Loch wird automatisch geöffnet und ist für alle verfügbar. Dass kann mit einer dynamischen Accessliste etwas veränder werden. Hier muss man ein Berechtigung haben, um eine Accessliste zu aktivieren. Das soll im Folgenden einmal durchgeführt werden.

20. Löschen Sie die konfigurierte Accessliste, deren Bindung ans Interface und die Zeitreferenz INFOSECTIME..

21. Erzeugen Sie einen Benutzeraccount auf dem Router für den User *infosec* mit dem Passwort **P@ssw0rd**.
    Wie lautet der Befehl?

    FW1(config)# username infosec password P@ssw0rd

22. Erzeugen Sie nun eine Accessliste, die eine statischen Accesslisteneintrag hat, der Telnet von PC1 zu FW1 zulässt und dann einen dynamischen Eintrag, der IP von PC1 zu PC2 für eine Minute zu lässt. Alles Andere soll von der Accessliste blockiert werden. Wie lautet die entsprechende Accessliste (fügen Sie wieder ein **log** an?

    access-list 101 permit tcp host 192.168.1.2 host 192.168.1.1 eq telnet
    access-list 101 dynamic testlist timeout 1 permit ip host 192.168.1.2 host 192.168.2.2 log

23. Weisen Sie die Accessliste dem Interface zu. Wie lauten die Befehle?

    interfase fa0/0
    ip access-group 101 in

24. Konfigurieren Sie nun den **autocommand** Befehl, der die Accessliste nach der Authentifizierung aktiviert und nach 1 Minute Inaktivität die Liste wieder deaktiviert. Wie lautet der Befehl?

    line vty 0 4
    login local
    autocommand access-enable host timeout 1

25. Starten Sie nun in einer DOS-Box ein Dauerping von PC1 zu PC2. Ist der Ping erfolgreich?

    Nein

26. Öffnen Sie auf PC1 eine weitere DOS-Box, starten Sie darin eine Telnetverbindung zu FW1 und loggen Sie sich mit ***infosec*** und ***P@ss0rd*** ein.

    Wie sieht es nun mit dem Ping aus? Ping ist erfolgreich!

27. Schauen Sie sich die Ausgabe des Log-Parameters der Accessliste auf FW1 an, wann wurde der erste Ping erlaubt. Geben Sie die Zeit an:

    10:36:25

28. Warten Sie, bis der Ping nicht mehr erfolgreich ist. Ab wann ist das? **10:37:25 (ca.)**

29. Löschen Sie wieder die Accessliste und alle zugehörigen Befehl. Jetzt sollen Sie eine reflexive Accessliste konfigurieren. Dabei ist es für ein Paket nur dann möglich von außen durch die Firewall zu kommen, wenn zuerst eine Verbindung von Innen gestartet wurde und darauf Antwortpakete gesendet werden. Im Beispiel sollen Sie die Accessliste so konfigurieren, dass Sie von PC2 zwar PC1 pingen können, aber nicht umgekehrt. Wie lauten die benötigten Befehle (Verwenden Sie wieder den **log** Parameter, prüfen Sie mit dem Fragezeichen *?,* an welche Stelle das log kommen muss!)?

    ip access-list extended OUTBOUNDFILTERS
    permit icmp host 192.168.2.2 host 192.168.1.2 log reflect ICMPTRAFFIC

    ip access-list extended INBOUNDFILTERS
    evaluate ICMPTRAFFIC

    interface fa0/0
    ip access-group INBOUNDFILTERS in
    ip access-group OUTBOUNDFILTERS out

30. Pingen Sie PC2 von PC1 aus. Ist der Ping erfolgreich? **Nein**

31. Pingen Sie PC1 von PC2 aus. Ist der Ping erfolgreich? **Ja**

Damit haben Sie alle vorgestellten Accesslisten einmal ausprobiert und gesehen, wir sie funktionieren. Löschen Sie alle Router mit dem Befehl **erase start.**

Es gibt noch weitere Accesslistentypen, die noch detailliertere Einstellungen zulassen. Diese Arten werden Ihnen, falls Sie noch ein Informatik-Masterstudium an der FH Aachen machen, im Fach ***Sicherheit in Datennetzen*** begegnen.