IP-toegangslijsten configureren en filteren

Inhoud

Inleiding

Dit document beschrijft verschillende typen IP-toegangscontrolelijsten (ACL's) en hoe deze netwerkverkeer kunnen filteren.

Voorwaarden

Vereisten

Er zijn geen specifieke voorwaarden van toepassing op dit document. De beschreven concepten zijn aanwezig in Cisco IOS® softwarereleases 8.3 en hoger. Dit wordt aangegeven onder elke ACL-functie.

Gebruikte componenten

In dit document worden verschillende typen ACL's beschreven. Sommige hiervan zijn aanwezig in Cisco IOS-softwarereleases 8.3, andere zijn in hogere softwarereleases vrijgegeven. Dit wordt aangegeven bij de beschrijving van elk type.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Conventies

Raadpleeg Cisco Technical Tips Conventies voor meer informatie over documentconventies.

Achtergrondinformatie

In dit document wordt beschreven hoe IP-toegangscontrolelijsten (ACL's) netwerkverkeer kunnen filteren. Er worden ook beknopte beschrijvingen gegeven van de IP ACL-typen, de functiebeschikbaarheid en een voorbeeld van het gebruik ervan in een netwerk.

> **Opmerking:** RFC 1700 bevat toegewezen nummers van bekende poorten.RFC 1918 bevat adrestoewijzing voor privaat internet, IP-adressen die normaal gesproken niet op het internet te zien zijn.

> **Opmerking:** Alleen geregistreerde Cisco-gebruikers kunnen interne informatie benaderen.

> **Opmerking:** ACL's kunnen ook worden gebruikt om verkeer naar netwerkadresomzetting (NAT) te definiëren, niet-IP-protocollen zoals AppleTalk of IPX te versleutelen of te filteren. Een beschrijving van deze functies valt buiten de reikwijdte van dit document.

ACL-concepten

Maskers

De maskers worden met IP adressen in IP ACLs gebruikt om te specificeren wat moet worden toegelaten en worden ontkend. Maskers voor het configureren van IP-adressen op interfaces starten met 255 en hebben de grootste waarde aan de linkerzijde. Bijvoorbeeld: IP-adres 10.165.202.129 met masker 255.255.255.224. De maskers voor IP ACLs zijn het omgekeerde, bijvoorbeeld, masker 0.0.0.255. Dit wordt soms genoemd een omgekeerd masker of een vervangingsmasker. Wanneer de waarde van het masker is opgesplitst in binair getal (0s en 1s), bepalen de resultaten welke adresbits moeten worden overwogen wanneer verkeer wordt verwerkt. Een 0 geeft aan dat de adresbits moeten worden meegenomen (exacte overeenkomst); a 1 in het masker is een *niet geven*. De onderstaande tabel maakt het concept duidelijker.

	Voorbeeld van masker
netwerkadres (verkeer dat moet worden verwerkt)	*10.1.1.0*
masker	*0.0.0.255*
netwerkadres (binair)	*00001010.00000001.00000001.00000000*
masker (binair)	*00000000.00000000.00000000.11111111*

Aan de hand van het binaire masker kunt u zien dat de eerste drie sets (octetten) exact moeten overeenkomen met het binaire netwerkadres (00001010.00000001.00000001). De laatste reeks getallen *geeft niet* (.11111111). Daarom *maakt* al het verkeer dat met 10.1.1 begint *niets uit* sinds het laatste octet. Met dit masker worden dus netwerkadres 10.1.1.1 tot en met 10.1.1.255 (10.1.1.x) verwerkt.

Trek het normale masker af van 255.255.255.255 om het inverse masker voor de ACL te bepalen. In dit voorbeeld wordt het inverse masker voor netwerkadres 172.16.1.0 bepaald met een normaal masker 255.255.255.0.

- 255.255.255.255 – 255.255.255.0 (normaal masker) = 0.0.0.255 (omgekeerd masker)

Let op de ACL-equivalenten.

- De bron/jokerteken van 0.0.0.0/255.255.255.255 betekent **elk.**
- De bron/wildcard van 10.1.1.2/0.0.0.0 is hetzelfde als **host 10.1.1.2**.

ACL-samenvatting

> **Opmerking:** Subnetmaskers kunnen ook worden aangegeven in een notatie met vaste lengte. 192.168.10.0/24 staat bijvoorbeeld voor 192.168.10.0 255.255.255.0.

Deze lijst toont hoe een reeks netwerken voor ACL-optimalisatie in één netwerk kan worden samengevat. We gaan uit van deze netwerken.

```
192.168.32.0/24
192.168.33.0/24
192.168.34.0/24
192.168.35.0/24
192.168.36.0/24
192.168.37.0/24
192.168.38.0/24
192.168.39.0/24
```

De eerste twee octetten en het laatste octet zijn gelijk voor elk netwerk. Via deze tabel wordt

getoond hoe deze in één netwerk kunnen worden samengevat.

Het derde octet voor de vorige netwerken kan worden geschreven zoals in deze tabel te zien is, correspondent voor de octet bit positie en adreswaarde voor elke bit.

Decimaal	128	64	32	16	8	4	2	1
32	0	0	1	0	0	0	0	0
33	0	0	1	0	0	0	0	1
34	0	0	1	0	0	0	1	0
35	0	0	1	0	0	0	1	1
36	0	0	1	0	0	1	0	0
37	0	0	1	0	0	1	0	1
38	0	0	1	0	0	1	1	0
39	0	0	1	0	0	1	1	1
	M	M	M	M	M	D	D	D

Aangezien de eerste vijf bits overeenkomen, kunnen de vorige acht netwerken in één netwerk worden samengevat (192.168.32.0/21 of 192.168.32.0 255.255.248.0). Alle acht mogelijke combinaties van de drie lage bits zijn relevant voor de betreffende netwerkreeksen. Met deze opdracht wordt een ACL gedefinieerd die dit netwerk toestaat. Als u 255.255.248.0 (normaal masker) aftrekt van 255.255.255.255, resulteert dat in 0.0.7.255.

```
access-list acl_permit permit ip 192.168.32.0 0.0.7.255
```

Bekijk deze set netwerken voor verdere uitleg.

```
192.168.146.0/24
192.168.147.0/24
192.168.148.0/24
192.168.149.0/24
```

De eerste twee octetten en het laatste octet zijn gelijk voor elk netwerk. Via deze tabel wordt getoond hoe deze kunnen worden samengevat.

Het derde octet voor de vorige netwerken kan worden geschreven zoals in deze tabel te zien is, correspondent voor de octet bit positie en adreswaarde voor elke bit.

Decimaal	128	64	32	16	8	4	2	1
146	1	0	0	1	0	0	1	0
147	1	0	0	1	0	0	1	1
148	1	0	0	1	0	1	0	0
149	1	0	0	1	0	1	0	1
	M	M	M	M	M	?	?	?

In tegenstelling tot het vorige voorbeeld kunnen deze netwerken niet in één netwerk worden samengevat. Als deze in één netwerk worden samengevat, worden ze 192.168.144.0/21 omdat het derde octet vijf vergelijkbare bits bevat. Dit samengevatte netwerk 192.168.144.0/21 bestrijkt een scala van netwerken van 192.168.144.0 tot 192.168.151.0. Van deze netwerken zijn 192.168.144.0, 192.168.145.0, 192.168.150.0 en 192.168.151.0 niet opgenomen in de lijst van vier netwerken . Om die specifieke netwerken op te nemen, zijn minimaal twee samengevatte netwerken nodig. De betreffende vier netwerken kunnen in deze twee netwerken worden samengevat:

- Voor netwerken 192.168.146.x en 192.168.147.x, komen alle bits overeen behalve de laatste,

die een *onverschilligheid* is. Dit kan worden geschreven als 192.168.146.0/23 (of 192.168.146.0 255.255.254.0).
- Voor netwerken 192.168.148.x en 192.168.149.x, komen alle bits overeen behalve de laatste, die een *onverschilligheid* is. Dit kan worden geschreven als 192.168.148.0/23 (of 192.168.148.0 255.255.254.0).

Deze uitvoer definieert een samengevatte ACL voor de eerdere netwerken.

```
!--- This command is used to allow access access for devices with IP
!--- addresses in the range from 192.168.146.0 to 192.168.147.254. access-list 10 permit
192.168.146.0 0.0.1.255
```

```
!--- This command is used to allow access access for devices with IP
!--- addresses in the range from 192.168.148.0 to 192.168.149.254 access-list 10 permit
192.168.148.0 0.0.1.255
```

ACL's verwerken

Verkeer dat de router binnenkomt, wordt vergeleken met ACL-vermeldingen op basis van de volgorde waarop de vermeldingen aankomen bij de router. Nieuwe vermeldingen worden aan het einde van de lijst toegevoegd. De router blijft controleren tot een overeenkomst wordt gevonden. Als er geen overeenkomst is gevonden wanneer de router het einde van de lijst bereikt, wordt het verkeer geweigerd. Om deze reden, moet u de vaak hit ingangen bovenaan de lijst hebben. Er is sprake van geïmpliceerde weigering van verkeer dat niet wordt toegestaan. Een single-entry ACL met slechts één deny ingang kan al verkeer ontkennen. Een ACL moet ten minste één vermelding bevatten die wordt toegestaan, anders wordt al het verkeer geblokkeerd. Deze twee ACL's (101 en 102) hebben hetzelfde effect.

```
!--- This command is used to permit IP traffic from 10.1.1.0
!--- network to 172.16.1.0 network. All packets with a source
!--- address not in this range will be rejected. access-list 101 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
```

```
!--- This command is used to permit IP traffic from 10.1.1.0
!--- network to 172.16.1.0 network. All packets with a source
!--- address not in this range will be rejected. access-list 102 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
access-list 102 deny ip any any
```

In het volgende voorbeeld volstaat de laatste vermelding. U hebt de eerste drie vermeldingen niet nodig omdat IP TCP, User Datagram Protocol (UDP) en Internet Control Message Protocol (ICMP) bevat.

```
!--- This command is used to permit Telnet traffic
!--- from machine 10.1.1.2 to machine 172.16.1.1. access-list 101 permit tcp host 10.1.1.2 host
172.16.1.1 eq telnet
```

```
!--- This command is used to permit tcp traffic from
!--- 10.1.1.2 host machine to 172.16.1.1 host machine. access-list 101 permit tcp host 10.1.1.2
host 172.16.1.1
```

```
!--- This command is used to permit udp traffic from
!--- 10.1.1.2 host machine to 172.16.1.1 host machine. access-list 101 permit udp host 10.1.1.2
host 172.16.1.1


!--- This command is used to permit ip traffic from
!--- 10.1.1.0 network to 172.16.1.10 network. access-list 101 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
```

Poorten en berichttypen definiëren

U kunt niet alleen ACL-bron en -bestemming definiëren, maar u kunt ook poorten, ICMP-berichttypen en andere parameters definiëren. Een goede bron van informatie voor bekende poorten is RFC 1700 . ICMP-berichttypen worden beschreven in RFC 792 .

De router kan beschrijvende tekst weergeven op enkele bekende poorten. Gebruik een**?**voor hulp.

```
access-list 102 permit tcp host 10.1.1.1 host 172.16.1.1 eq ?
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
```

Tijdens de configuratie zet de router numerieke waarden tevens om naar meer gebruiksvriendelijke waarden. Dit is een voorbeeld waar u het ICMP berichttype nummer typt, en het veroorzaakt de router om het aantal in een naam om te zetten.

```
access-list 102 permit icmp host 10.1.1.1 host 172.16.1.1 14
```

wordt

```
access-list 102 permit icmp host 10.1.1.1 host 172.16.1.1 timestamp-reply
```

ACL's toepassen

U kunt ACL's definiëren en deze nog steeds niet toepassen. Maar de ACL's zijn pas van kracht wanneer deze op de interface van de router worden toegepast. Het is raadzaam om de ACL toe te passen op de interface die zich het dichtst bij de bron van het verkeer bevindt. Zoals in dit voorbeeld wordt getoond, wanneer u probeert verkeer van bron naar bestemming te blokkeren, kunt u een inkomende ACL op E0 op router A toepassen in plaats van een uitgaande lijst op E1 op router C. Een toegangslijst heeft **adeny ip om het even welk** impliciet aan het eind van om het even welke toegang-lijst. Als het verkeer aan een DHCP-verzoek is gerelateerd en als dit niet expliciet is toegestaan, wordt het verkeer verbroken omdat als u DHCP-verzoek in IP bekijkt, het bronadres s=0.0.0.0 (Ethernet1/0), d=255.255.255.255, len 604, rcvd 2 UDP src=68, dst=67 is. Het bronIP-adres is 0.0.0 en het doeladres is 255.255.255. 8 en bestemming 67. Daarom moet u dit soort verkeer in uw access-list toestaan of anders wordt het verkeer laten vallen wegens impliciet ontkennen aan het eind van de verklaring.

> **Opmerking:** Voor UDP-verkeer dat moet worden doorgegeven, moet UDP-verkeer ook expliciet door de ACL worden toegestaan.

In, uit, inkomend, uitgaand, bron en bestemming definiëren

De router gebruikt de begrippen in, uit, bron en bestemming als verwijzingen. Verkeer op de router kan worden vergeleken met verkeer op de snelweg. Als je een politieagent was in Pennsylvania en een vrachtwagen wilde stoppen die van Maryland naar New York reist, dan is de bron van de vrachtwagen Maryland, en de bestemming van de vrachtwagen is New York. De wegversperring zou kunnen worden toegepast aan de grens tussen Pennsylvania en New York (out) of de grens tussen Maryland en Pennsylvania (in).

Bij een router hebben deze begrippen dezelfde betekenis.

- **Uit – verkeer dat al door de router is geleid en de interface verlaat.** De bron is waar het verkeer is geweest, aan de andere kant van de router, en de bestemming is waar het verkeer naartoe gaat.
- **In – verkeer dat bij de interface aankomt en vervolgens door de router wordt geleid.** De bron is waar het verkeer is geweest en de bestemming is waar het verkeer naartoe gaat, aan de andere kant van de router.
- **Inkomend – als de ACL inkomend is en de router een pakket ontvangt, worden in de Cisco IOS-software de criteriavermeldingen in de ACL gecontroleerd op overeenkomsten.** Als het pakket wordt toegestaan, verwerkt de software het pakket. Als het pakket wordt geweigerd, negeert de software het pakket.
- **Uitgaand – als de ACL uitgaand is en een pakket via de software wordt gerouteerd naar de uitgaande interface, worden in de Cisco IOS-software de criteriavermeldingen in de ACL gecontroleerd op overeenkomsten.** Als het pakket wordt toegestaan, verzendt de software het pakket. Als het pakket wordt geweigerd, negeert de software het pakket.

De in-ACL heeft een bron op een segment van de interface waar deze wordt toegepast, en een bestemming op een andere interface. De uit-ACL heeft een bron op een segment op een andere interface dan die waarop deze wordt toegepast, en een bestemming op de interface waar deze wordt toegepast.

ACL's bewerken

Het bewerken van een ACL vereist speciale aandacht. Als u bijvoorbeeld een specifieke regel wilt verwijderen uit een bestaande genummerde ACL, zoals hier getoond, wordt de gehele ACL verwijderd.

```
!--- The access-list 101 denies icmp from any to any network
!--- but permits IP traffic from any to any network. router#configure terminal
  Enter configuration commands, one per line.  End with CNTL/Z.
  router(config)#access-list 101 deny icmp any any
  router(config)#access-list 101 permit ip any any
  router(config)#^Z

  router#show access-list
  Extended IP access list 101
```

```
      deny icmp any any
      permit ip any any
  router#
  *Mar  9 00:43:12.784: %SYS-5-CONFIG_I: Configured from console by console

  router#configure terminal
  Enter configuration commands, one per line.  End with CNTL/Z.
  router(config)#no access-list 101 deny icmp any any
  router(config)#^Z

  router#show access-list
  router#
  *Mar  9 00:43:29.832: %SYS-5-CONFIG_I: Configured from console by console
```

Kopieer de configuratie van de router naar een TFTP-server of een tekstverwerker (zoals Notepad) om genummerde ACL's te bewerken. Voer de wijzigingen door en kopieer de configuratie weer terug naar de router.

U kunt ook als volgt te werk gaan.

```
router#configure terminal
  Enter configuration commands, one per line.
  router(config)#ip access-list extended test

!--- Permits IP traffic from 10.2.2.2 host machine to 10.3.3.3 host machine. router(config-ext-
nacl)#permit ip host 10.2.2.2 host 10.3.3.3

!--- Permits www traffic from 10.1.1.1 host machine to 10.5.5.5 host machine. router(config-ext-
nacl)#permit tcp host 10.1.1.1 host 10.5.5.5 eq www

!--- Permits icmp traffic from any to any network. router(config-ext-nacl)#permit icmp any any

!--- Permits dns traffic from 10.6.6.6 host machine to 10.10.10.0 network. router(config-ext-
nacl)#permit udp host 10.6.6.6 10.10.10.0 0.0.0.255 eq domain
  router(config-ext-nacl)#^Z
  1d00h: %SYS-5-CONFIG_I: Configured from console by consoles-l

  router#show access-list
  Extended IP access list test
      permit ip host 10.2.2.2 host 10.3.3.3
      permit tcp host 10.1.1.1 host 10.5.5.5 eq www
      permit icmp any any
      permit udp host 10.6.6.6 10.10.10.0 0.0.0.255 eq domain
```

Alle verwijderingen worden uit de ACL gehaald; alle toevoegingen worden aan het einde van de ACL toegevoegd.

```
router#configure terminal
   Enter configuration commands, one per line.  End with CNTL/Z.
   router(config)#ip access-list extended test

!--- ACL entry deleted. router(config-ext-nacl)#no permit icmp any any

!--- ACL entry added. router(config-ext-nacl)#permit gre host 10.4.4.4 host 10.8.8.8
   router(config-ext-nacl)#^Z
   1d00h: %SYS-5-CONFIG_I: Configured from console by consoles-l

   router#show access-list
   Extended IP access list test
       permit ip host 10.2.2.2 host 10.3.3.3
       permit tcp host 10.1.1.1 host 10.5.5.5 eq www
```

```
       permit udp host 10.6.6.6 10.10.10.0 0.0.0.255 eq domain
       permit gre host 10.4.4.4 host 10.8.8.8
```

U kunt in Cisco IOS ook ACL-regels toevoegen aan genummerde standaard of uitgebreide ACL's op volgnummer. Hieronder volgt een voorbeeld van de configuratie:

Configureer de uitgebreide ACL als volgt:

```
Router(config)#access-list 101 permit tcp any any
Router(config)#access-list 101 permit udp any any
Router(config)#access-list 101 permit icmp any any
Router(config)#exit
Router#
```

Geef deze manier van toegang-lijst bevel uit om de ACL ingangen te bekijken. De volgnummers, zoals 10, 20 en 30, worden hier ook weergegeven.

```
Router#show access-list
Extended IP access list 101
    10 permit tcp any any
    20 permit udp any any
    30 permit icmp any any
```

Voer de vermelding voor ACL 101 toe met volgnummer 5.

Voorbeeld 1:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list extended 101
Router(config-ext-nacl)#5 deny tcp any any eq telnet
Router(config-ext-nacl)#exit
Router(config)#exit
Router#
```

In de output van toegangslijsten wordt volgnummer 5 ACL toegevoegd als eerste ingang van de toegangslijst 101.

```
Router#show access-list
Extended IP access list 101
    5 deny tcp any any eq telnet
    10 permit tcp any any
    20 permit udp any any
    30 permit icmp any any
Router#
```

Voorbeeld 2:

```
internetrouter#show access-lists
Extended IP access list 101
    10 permit tcp any any
    15 permit tcp any host 172.16.2.9
    20 permit udp host 172.16.1.21 any
    30 permit udp host 172.16.1.22 any

internetrouter#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
internetrouter(config)#ip access-list extended 101
internetrouter(config-ext-nacl)#18 per tcp any host 172.16.2.11
```

```
internetrouter(config-ext-nacl)#^Z

internetrouter#show access-lists
Extended IP access list 101
    10 permit tcp any any
    15 permit tcp any host 172.16.2.9
    18 permit tcp any host 172.16.2.11
    20 permit udp host 172.16.1.21 any
    30 permit udp host 172.16.1.22 any
internetrouter#
```

U kunt de standaard ACL op dezelfde manier configureren:

```
internetrouter(config)#access-list 2 permit 172.16.1.2
internetrouter(config)#access-list 2 permit 172.16.1.10
internetrouter(config)#access-list 2 permit 172.16.1.11

internetrouter#show access-lists
Standard IP access list 2
    30 permit 172.16.1.11
    20 permit 172.16.1.10
    10 permit 172.16.1.2

internetrouter(config)#ip access-list standard 2
internetrouter(config-std-nacl)#25 per 172.16.1.7
internetrouter(config-std-nacl)#15 per 172.16.1.16

internetrouter#show access-lists
Standard IP access list 2
    15 permit 172.16.1.16
    30 permit 172.16.1.11
    20 permit 172.16.1.10
    25 permit 172.16.1.7
    10 permit 172.16.1.2
```

Het belangrijkste verschil in een standaard toegangslijst is dat Cisco IOS een vermelding in afstammende volgorde van het IP-adres toevoegt, niet op een volgnummer.

In dit voorbeeld worden de verschillende vermeldingen getoond, bijvoorbeeld hoe u een IP-adres (192.168.100.0) of de netwerken (10.10.10.0) kunt toestaan.

```
internetrouter#show access-lists
Standard IP access list 19
    10 permit 192.168.100.0
    15 permit 10.10.10.0, wildcard bits 0.0.0.255
    19 permit 10.101.110.0, wildcard bits 0.0.0.255
    25 deny any
```

Voeg de vermelding aan ACL 2 toe om IP-adres 172.22.1.1 toe te staan:

```
internetrouter(config)#ip access-list standard 2
internetrouter(config-std-nacl)#18 permit 172.22.1.1
```

Deze vermelding wordt boven aan de lijst toegevoegd om prioriteit te geven aan het specifieke IP-adres in plaats van aan het netwerk.

```
internetrouter#show access-lists
Standard IP access list 19
```

```
10 permit 192.168.100.0
18 permit 172.22.1.1
15 permit 10.10.10.0, wildcard bits 0.0.0.255
19 permit 10.101.110.0, wildcard bits 0.0.0.255
25 deny   any
```

Opmerking: De vorige ACL's worden niet ondersteund in security applicaties zoals de ASA-/PIX-firewall.

Richtlijnen voor het wijzigen van ACL's wanneer deze worden toegepast op crypto maps

- Als u aan een huidige configuratie van de toegangslijst toevoegt, is het niet nodig de crypto-kaart te verwijderen. U mag deze rechtstreeks toevoegen zonder de crypto map te verwijderen: dit wordt ondersteund en is acceptabel.
- Als u toegang-lijst ingang van een huidige toegang-lijsten moet wijzigen of schrappen, dan moet u de crypto kaart uit de interface verwijderen. Nadat u de crypto map heeft verwijderd, kunt u de wijzigingen doorvoeren in de ACL en de crypto map weer toevoegen. Wijzigingen zoals verwijdering van de ACL die u doorvoert zonder de crypto map te verwijderen, worden niet ondersteund. Dit kan leiden tot onvoorspelbaar gedrag.

Problemen oplossen

Hoe verwijder ik een ACL uit een interface?

Activeer de configuratiemodus en geef **no op vóór de opdracht access-group (zie voorbeeld) om een ACL uit een interface te verwijderen.**

```
interface <interface-name> no ip access-group <acl-number> {in|out}
```

Wat moet ik doen als te veel verkeer wordt geweigerd?

Als te veel verkeer wordt geweigerd, moet u de logica van de ACL nagaan. U kunt ook een aanvullende, bredere lijst definiëren en toepassen. Met de opdracht **show ip access-lists wordt het aantal pakketten getoond en kunt u zien welke ACL-vermelding wordt gebruikt.** Met het trefwoord log aan het einde van de afzonderlijke ACL-vermeldingen worden het ACL-nummer en poortspecifieke informatie getoond, en wordt aangegeven of het pakket was toegestaan/geweigerd.

Opmerking: Het trefwoord log-input is beschikbaar in Cisco IOS-softwarerelease 11.2 en hoger, en in bepaalde op Cisco IOS-softwarerelease 11.1 gebaseerde software die specifiek voor de serviceprovidermarkt is gemaakt. In oudere softwarereleases wordt dit trefwoord niet ondersteund. Het gebruik van dit trefwoord omvat de opgegeven interface en het MAC-adres van de bron, waar van toepassing.

Hoe kan ik fouten opsporen op pakketniveau bij gebruik van een Cisco-router?

Via de onderstaande procedure wordt het foutopsporingsproces uitgelegd. Zorg ervoor dat er geen momenteel toegepaste ACL's zijn, dat er een ACL is en dat fast switching is ingeschakeld.

Opmerking: Ga zeer zorgvuldig te werk wanneer u fouten opspoort op een systeem met veel verkeer. Gebruik een ACL om fouten in specifiek verkeer op te sporen. Maar zorg voor het proces en de verkeersstroom.

1. Gebruik **de** opdracht **access** list om de gewenste gegevens op te nemen.In dit voorbeeld is gegevensvastlegging ingesteld voor het bestemmingsadres 10.2.6.6 van bronadres 10.2.6.6.
   ```
   access-list 101 permit ip any host 10.2.6.6
   access-list 101 permit ip host 10.2.6.6 any
   ```

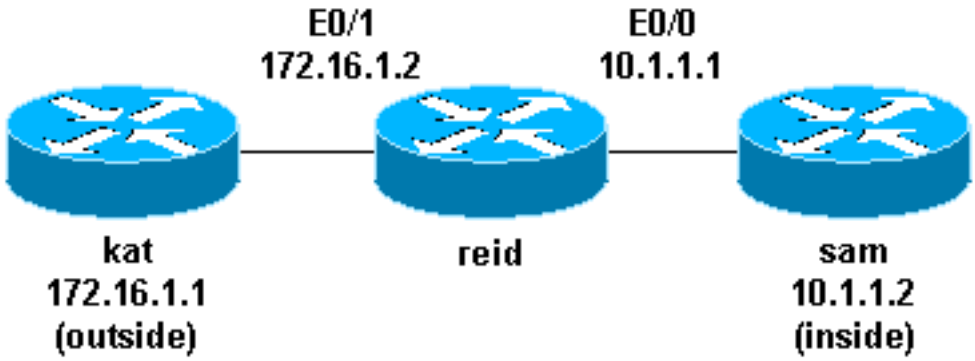
2. Schakel fast switching uit op de betrokken interfaces. U ziet alleen het eerste pakket als fast switching niet is uitgeschakeld.
   ```
   configure terminal
   interface
   ```

3. Voer de opdracht **terminal monitor** uit in de enable-modus om de uitvoer van de opdracht debug en systeemfoutmeldingen voor de huidige terminal en sessie te tonen.
4. Gebruik het **debug ip-pakket 101** of **debug ip-pakket 101-**detailopdracht om het debug-proces te starten.
5. Voer de opdracht **no debug all uit in de enable-modus en de opdracht interface configuration om het foutopsporingsproces te stoppen.**
6. Start cacheopslag opnieuw op.
   ```
   configure terminal
   interface
   ```

De verschillende typen IP ACL's

In deze sectie worden ACL-typen beschreven.

Netwerkdiagram



Standaard ACL's

Standaard ACL's zijn het oudste type ACL. Ze gaan terug naar Cisco IOS-softwarerelease 8.3. Standaard ACL's beheren verkeer door het bronadres van de IP-pakketten te vergelijken met de in de ACL geconfigureerde adressen.

Hieronder staat de opdrachtsyntaxis van een standaard ACL.

```
access-list <access-list-number> {permit|deny} {host|source source-wildcard|any}
```

In alle softwarereleases kan het *toegangslijst-nummer* van 1 tot 99 *zijn*. In Cisco IOS-softwarerelease 12.0.1 beginnen standaard ACL's extra nummers (1300 tot 1999) te gebruiken. Deze aanvullende nummers worden aangeduid als uitgebreide IP ACL's. In Cisco IOS-softwarerelease 11.2 is de mogelijkheid toegevoegd om access-list *name in standaard ACL's te gebruiken.*

Een *bron/bron-wildcard*-instelling van 0.0.0.0/255.255.255.255 kan als **elke** instelling worden gespecificeerd. Dit jokerteken kan worden weggelaten als het alleen nullen betreft. Daarom is host 10.1.1.2 0.0.0.0 hetzelfde als host 10.1.1.2.

Nadat u de ACL heeft gedefinieerd, moet deze op de interface worden toegepast (inkomend of uitgaand). In eerdere softwarereleases was 'out' de standaard wanneer geen trefwoord 'out' of 'in' was opgegeven. In latere softwarereleases moest de richting worden opgegeven.

```
interface <interface-name>
 ip access-group number {in|out}
```

Hierna volgt een voorbeeld van het gebruik van een standaard ACL om al het verkeer te blokkeren, met uitzondering van verkeer afkomstig van bron 10.1.1.x.

```
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 1 in
!
access-list 1 permit 10.1.1.0 0.0.0.255
```

Uitgebreide ACL's

Uitgebreide ACL's zijn geïntroduceerd in Cisco IOS-softwarerelease 8.3. Uitgebreid ACL-controleverkeer door de bron- en doeladressen van de IP-pakketten te vergelijken met de in de ACL geconfigureerde adressen.

Hieronder staat de opdrachtsyntaxis van uitgebreide ACL's. Lijnen zijn hier omwikkeld om ruimte-redenen.

IP

```
access-list access-list-number
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} protocol source source-wildcard destination destination-wildcard [precedence
precedence]
     [tos tos] [log|log-input] [time-range time-range-name]
```

ICMP

```
access-list access-list-number
```

```
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} icmp source source-wildcard destination destination-wildcard
     [icmp-type [icmp-code] |icmp-message] [precedence precedence] [tos tos] [log|log-input]
     [time-range time-range-name]
```

TCP

```
access-list access-list-number
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} tcp source source-wildcard [operator [port]]
     destination destination-wildcard [operator [port]]
     [established] [precedence precedence] [tos tos]
     [log|log-input] [time-range time-range-name]
```

UDP

```
access-list access-list-number
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} udp source source-wildcard [operator [port]]
     destination destination-wildcard [operator [port]]
     [precedence precedence] [tos tos] [log|log-input]
     [time-range time-range-name]
```

In alle softwarereleases kan het *toegangslijstnummer* 100 tot 199 zijn. In Cisco IOS-softwarerelease 12.0.1 beginnen uitgebreide ACL's extra nummers (2000 tot 2699) te gebruiken. Deze aanvullende nummers worden aangeduid als uitgebreide IP ACL's. In Cisco IOS-softwarerelease 11.2 is de mogelijkheid toegevoegd om access-list *name in uitgebreide ACL's te gebruiken.*

De waarde 0.0.0.0/255.255.255.255 kan worden opgegeven als **any (elk).** Nadat u de ACL heeft gedefinieerd, moet deze op de interface worden toegepast (inkomend of uitgaand). In eerdere softwarereleases was 'out' de standaard wanneer geen trefwoord 'out' of 'in' was opgegeven. In latere softwarereleases moest de richting worden opgegeven.

```
interface <interface-name>
 ip access-group {number|name} {in|out}
```

Dit uitgebreide ACL wordt gebruikt om verkeer op het 10.1.1.x-netwerk (binnen) toe te laten en om ping-reacties van de buitenkant te ontvangen terwijl het ongevraagde pings van mensen buiten voorkomt, wat al ander verkeer toestaat.

```
interface Ethernet0/1
 ip address 172.16.1.2 255.255.255.0
 ip access-group 101 in
!
access-list 101 deny icmp any 10.1.1.0 0.0.0.255 echo access-list 101 permit ip any 10.1.1.0
0.0.0.255
```

> **Opmerking:** Sommige toepassingen, zoals netwerkbeheer, vereisen pings voor een keepalive-functie. Als dit het geval is, kunt u inkomende pings beperken die worden geblokkeerd of korreliger zijn in toegelaten/ontkende IPs.

Lock and Key ACL's (dynamische ACL's)

Lock en key, ook bekend als dynamische ACL's, is geïntroduceerd in Cisco IOS-softwarerelease 11.1. Deze functie is afhankelijk van Telnet, verificatie (lokaal of extern) en uitgebreide ACL's.

Lock and Key-configuratie begint met de toepassing van een uitgebreide ACL om verkeer via de router te blokkeren. Gebruikers die de router willen passeren, worden door de uitgebreide ACL geblokkeerd totdat het Telnet-protocol is uitgevoerd en de gebruikers zijn geverifieerd. De Telnet-verbinding daalt dan en er wordt een enkelvoudige dynamische ACL toegevoegd aan de uitgebreide ACL die bestaat. Hierdoor wordt verkeer gedurende een bepaalde tijdsperiode toegestaan; inactiviteit en absolute time-outs zijn mogelijk.

Hieronder staat de opdrachtsyntaxis voor Lock and Key-configuratie met lokale verificatie.

```
username <user-name> password <password>
!
interface <interface-name>
 ip access-group {number|name} {in|out}
```

De ACL met één vermelding in deze opdracht wordt na verificatie dynamisch toegevoegd aan de bestaande ACL.

```
access-list access-list-number dynamic name {permit|deny} [protocol]
{source source-wildcard|any} {destination destination-wildcard|any}
[precedence precedence][tos tos][established] [log|log-input]
[operator destination-port|destination port]

line vty <line_range>
login local
```

Hieronder staat een basisvoorbeeld van Lock and Key.

```
username test password 0 test

!--- Ten (minutes) is the idle timeout. username test autocommand access-enable host timeout 10
!
interface Ethernet0/0
  ip address 10.1.1.1 255.255.255.0
  ip access-group 101 in
!
access-list 101 permit tcp any host 10.1.1.1 eq telnet

!--- 15 (minutes) is the absolute timeout. access-list 101 dynamic testlist timeout 15 permit ip
10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
!
line vty 0 4
 login local
```

Nadat de gebruiker bij 10.1.1.2 een Telnet-verbinding naar 10.1.1.1 tot stand heeft gebracht, wordt de dynamische ACL toegepast. Daarna wordt de verbinding verbroken en kan de gebruiker naar het netwerk 172.16.1.x.

ACL's met benoemde IP's

IP met de naam ACL's is geïntroduceerd in Cisco IOS-softwarerelease 11.2. Hierdoor kunnen standaard en uitgebreide ACL's in plaats van nummers worden gegeven.

Hieronder staat de opdrachtsyntaxis van ACL's met benoemde IP's.

```
ip access-list {extended|standard} name
```

Hieronder staat een TCP-voorbeeld:

```
{permit|deny} tcp source source-wildcard [operator [port]] destination destination-wildcard
[operator [port]] [established] [precedence precedence] [tos tos] [log] [time-range time-range-
name]
```

Hierna volgt een voorbeeld van het gebruik van een benoemde ACL om al het verkeer te blokkeren met uitzondering van de Telnet-verbinding van host 10.1.1.2 naar host 172.16.1.1.

```
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group in_to_out in
!
ip access-list extended in_to_out
 permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
```

Reflexieve ACL's

Reflexieve ACL's zijn geïntroduceerd in Cisco IOS-softwarerelease 11.3. Reflexieve ACL's maken het mogelijk IP-pakketten te filteren op basis van sessieinformatie van de bovenste laag. Deze ACL's worden doorgaans gebruikt om uitgaand verkeer toe te staan en inkomend verkeer te beperken als reactie op sessies die plaatsvinden in de router.

Reflexieve ACL's kunnen alleen worden gedefinieerd met uitgebreide benoemde IP ACL's. Ze kunnen niet worden gedefinieerd met genummerde of standaard benoemde IP ACL's of andere protocol ACL's. Reflexieve ACL's kunnen worden gebruikt in combinatie met andere standaard en statische uitgebreide ACL's.

Hieronder staat de syntaxis voor diverse reflexieve ACL-opdrachten.

```
interface <interface-name>
 ip access-group {number|name} {in|out}
!
ip access-list extended <name>
 permit protocol any any reflect name [timeoutseconds]
!
ip access-list extended <name>
 evaluate <name>
```

Dit is een voorbeeld van de vergunning van uitgaand en binnenkomend ICMP-verkeer, terwijl het alleen TCP-verkeer toestaat dat van binnenuit geïnitieerd is, wordt ander verkeer geweigerd.

```
ip reflexive-list timeout 120
```

```
!
interface Ethernet0/1
 ip address 172.16.1.2 255.255.255.0
 ip access-group inboundfilters in
 ip access-group outboundfilters out
!
ip access-list extended inboundfilters
 permit icmp 172.16.1.0 0.0.0.255 10.1.1.0 0.0.0.255
 evaluate tcptraffic

!--- This ties the reflexive ACL part of the outboundfilters ACL,
!--- called tcptraffic, to the inboundfilters ACL. ip access-list extended outboundfilters
 permit icmp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255
 permit tcp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 reflect tcptraffic
```

Tijdgebaseerde ACL's met tijdbereiken

Tijdgebaseerde ACL's werden geïntroduceerd in Cisco IOS-softwarerelease 12.0.1.T. Hoewel deze qua functionering vergelijkbaar werken als uitgebreide ACL's, is hierbij tijdgebaseerde toegangscontrole mogelijk. Er wordt een tijdbereik gecreëerd dat specifieke tijden van de dag of week definieert om tijdgebaseerde ACL's te implementeren. Het tijdbereik wordt gedefinieerd op naam en vervolgens wordt er via een functie naar verwezen. De tijdrestricties worden dan op de functie zelf toegepast. Het tijdbereik maakt gebruik van de systeemklok van de router. De routerklok kan worden gebruikt, maar de functie werkt het beste met NTP-synchronisatie (Network Time Protocol).

Hieronder staan tijdgebaseerde ACL-opdrachten.

```
!--- Defines a named time range. time-range time-range-name

!--- Defines the periodic times. periodic days-of-the-week hh:mm to [days-of-the-week] hh:mm

!--- Or, defines the absolute times. absolute [start time date] [end time date]

!--- The time range used in the actual ACL. ip access-list name|number time-rangename_of_time-
range
```

In dit voorbeeld is op maandag, woensdag en vrijdag tijdens kantooruren een Telnet-verbinding toegestaan van het interne naar het externe netwerk:

```
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 101 in
!
access-list 101 permit tcp 10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 eq telnet time-range
EVERYOTHERDAY
!
time-range EVERYOTHERDAY
 periodic Monday Wednesday Friday 8:00 to 17:00
```

Vermeldingen in IP ACL's met opmerking

Vermeldingen in IP ACL's met opmerking werden geïntroduceerd in Cisco IOS-softwarerelease 12.0.2.T. Opmerkingen kunnen worden opgenomen in standaard of uitgebreide IP ACL's en zorgen ervoor dat ACL's eenvoudiger te begrijpen zijn.

Hieronder staat de opdrachtsyntaxis van een benoemde IP ACL met opmerkingen.

```
ip access-list {standard|extended} <access-list-name> remark remark
```

Hieronder staat de opdrachtsyntaxis van een genummerde IP ACL met opmerkingen.

```
access-list <access-list-number> remark remark
```

Dit is een voorbeeld van opmerkingen binnen een genummerde ACL.

```
interface Ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 101 in
!
access-list 101 remark permit_telnet
access-list 101 permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
```

Contextgebaseerd toegangsbeheer

Contextgebaseerd toegangsbeheer (CBAC; Context-Based Access Control) werd geïntroduceerd in Cisco IOS-softwarerelease 12.0.5.T en vereist de Cisco IOS Firewall-functieset. CBAC controleert verkeer dat door de firewall loopt om statusinformatie van TCP- en UDP-sessies te detecteren en beheren. Deze statusinformatie wordt gebruikt om tijdelijke openingen in de ACL's van de firewall te maken. Configureer inspectielijsten in de richting van de stroom van verkeersinitiatie om terugkeerverkeer en extra gegevensverbindingen voor toelaatbare sessies toe te staan, sessies die afkomstig zijn uit het beveiligde interne netwerk.

Hieronder staat de syntaxis voor CBAC.

```
ip inspect name inspection-name protocol [timeoutseconds]
```

Hierna volgt een voorbeeld van het gebruik van CBAC om uitgaand verkeer te controleren. Uitgebreide ACL 111 blokkeert normaliter retourverkeer anders dan ICMP zonder CBAC-openingen voor het retourverkeer.

```
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http timeout 3600
ip inspect name myfw tcp timeout 3600
ip inspect name myfw udp timeout 3600
ip inspect name myfw tftp timeout 3600
! interface Ethernet0/1 ip address 172.16.1.2 255.255.255.0 ip access-group 111 in ip inspect
myfw out !
access-list 111 deny icmp any 10.1.1.0 0.0.0.255 echo access-list 111 permit icmp any 10.1.1.0
0.0.0.255
```

Verificatieproxy

Verificatieproxy werd geïntroduceerd in Cisco IOS-softwarerelease 12.0.5.T. Hiervoor is de Cisco IOS Firewall-functieset vereist. Verificatieproxy wordt gebruikt om inkomende of uitgaande gebruikers (of beide) te verifiëren. Gebruikers die normaliter worden geblokkeerd door een ACL,

kunnen een browser starten om de firewall te passeren en vervolgens te verifiëren via een TACACS+ of RADIUS-server. De server stuurt aanvullende ACL-vermeldingen naar de router zodat de gebruikers na verificatie toegang krijgen.

Verificatieproxy is vergelijkbaar met Lock and Key ACL's (dynamische ACL's). Er zijn echter enkele verschillen:

- Lock and Key wordt ingeschakeld door een Telnet-verbinding naar de router. Verificatieproxy wordt ingeschakeld door HTTP via de router.
- Verificatieproxy moet een externe server gebruiken.
- Verificatieproxy kan de toevoeging van meerdere dynamische lijsten verwerken. Lock and Key kan slechts één extra dynamische lijst verwerken.
- Verificatieproxy kent een absolute time-out maar geen time-out bij inactiviteit. Lock and Key kent beide.

Raadpleeg Cisco Secure Integrated Software Configuration Cookbook (Cisco's configuratiegids voor beveiligde geïntegreerde software) voor voorbeelden van verificatieproxy.

Turbo ACL's

Turbo ACL's werden geïntroduceerd in Cisco IOS-softwarerelease 12.1.5.T en zijn alleen beschikbaar op 7200-, 7500- en andere hoogwaardige platforms. De functie voor turbo ACL's is ontwikkeld om ACL's efficiënter te verwerken en zo de routerprestaties te verbeteren.

Gebruik de opdracht **access-list compiled voor turbo ACL's.** Dit is een voorbeeld van een gecompileerde ACL.

```
access-list 101 permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
access-list 101 permit tcp host 10.1.1.2 host 172.16.1.1 eq ftp
access-list 101 permit udp host 10.1.1.2 host 172.16.1.1 eq syslog
access-list 101 permit udp host 10.1.1.2 host 172.16.1.1 eq tftp
access-list 101 permit udp host 10.1.1.2 host 172.16.1.1 eq ntp
```

Nadat de standaard of uitgebreide ACL is gedefinieerd, gebruikt u de opdracht **global configuration om te compileren.**

```
!--- Tells the router to compile. access-list compiled
!
interface Ethernet0/1
 ip address 172.16.1.2 255.255.255.0

!--- Applies to the interface.  ip access-group 101 in
```

Met de opdracht **show access-list compiled worden statistieken van de ACL getoond.**

Gedistribueerde tijdgebaseerde ACL's

Gedistribueerde tijdgebaseerde ACL's werden geïntroduceerd in Cisco IOS-softwarerelease 12.2.2.T om tijdgebaseerde ACL's te implementeren op VPN-routers uit de 7500-Series. Voorafgaand aan de introductie van gedistribueerde tijdgebaseerde ACL's werden tijdgebaseerde ACL's niet ondersteund op lijnkaarten voor Cisco-routers uit de 7500-Series. Wanneer tijdgebaseerde ACL's werden geconfigureerd, gedroegen deze zich als normale ACL's. Wanneer een interface op een lijnkaart was geconfigureerd met tijdgebaseerde ACL's, werden de naar de

interface geswitchte pakketten niet via de lijnkaart gedistribueerd, maar ter verwerking doorgestuurd naar de routeprocessor.

De syntaxis voor gedistribueerde tijdgebaseerde ACL's is hetzelfde als voor tijdgebaseerde ACL's met toevoeging van de opdrachten met betrekking tot de status van de Inter Processor Communication (IPC)-berichten tussen de routeprocessor en de lijnkaart.

```
debug time-range ipc
show time-range ipc
clear time-range ipc
```

Ontvangst-ACL's

Ontvangst-ACL's worden gebruikt om de beveiliging op Cisco 12000-routers te verhogen door de gigabit routeprocessor (GRP) van de router te beschermen tegen onnodig en potentieel schadelijk verkeer. Ontvangst-ACL's werden toegevoegd als een speciale uitzondering op de onderhoudsbeperking voor Cisco IOS-softwarerelease 12.0.21S2 en geïntegreerd in 12.0(22)S. Raadpleeg GSR: Ontvang toegangscontrolelijsten voor meer informatie.

ACL's met infrastructuurbescherming

De ACL's van de infrastructuur worden gebruikt om het risico en de doeltreffendheid van directe infrastructuuraanvallen te minimaliseren door de uitdrukkelijke toestemming van alleen bevoegd verkeer aan de infrastructuurapparatuur terwijl het al ander transitoverkeer toestaat. Raadpleeg Protecting Your Core: Infrastructure Protection Access Control Lists (Uw core beschermen: toegangscontrolelijsten voor infrastructuurbescherming) voor meer informatie.

Doorgifte-ACL's

Doorgifte-ACL's worden gebruikt om de netwerkbeveiliging te verhogen, aangezien deze uitsluitend vereist verkeer toelaten tot uw netwerk of netwerken. Raadpleeg Transit Access Control Lists: Filtering at Your Edge (Doorgifte-toegangscontrolelijsten: filteren aan de edge) voor meer informatie.

Gerelateerde informatie

- Veelgebruikte IP ACL's configureren
- RFC 1700
- RFC 1918
- Ondersteuningspagina voor ACL's
- Cisco IOS Firewall
- Technische ondersteuning en documentatie – Cisco Systems

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.