Tutorial 2 AC Lijsten

Toegangsbeheerlijsten (ACL's) bieden een manier om pakketten te filteren door een gebruiker toe te staan toe te staan of te weigeren dat IP-pakketten bepaalde interfaces kruisen. Stel je voor dat je naar een beurs komt en de bewaker tickets ziet controleren. Hij laat alleen mensen met geschikte tickets toe. Welnu, de functie van een toegangslijst is hetzelfde als die voogd.

Toegangslijsten filteren netwerkverkeer door te bepalen of pakketten worden doorgestuurd of geblokkeerd op de interfaces van de router op basis van de criteria die u hebt opgegeven in de toegangslijst.

Als u ACL's wilt gebruiken, moet de systeembeheerder eerst ACL's configureren en deze vervolgens toepassen op specifieke interfaces. Er zijn 3 populaire typen ACL's: Standard, Extended en Named ACL's.

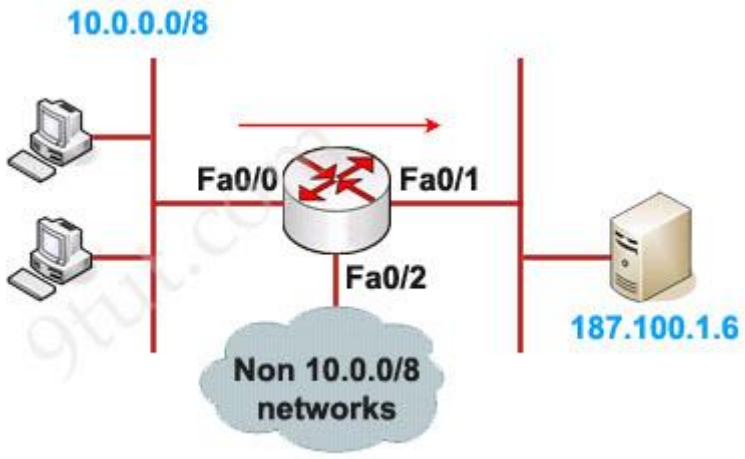
Standard IP Access List

Standaard IP-lijsten (1-99) controleren alleen de bronadressen van alle IP-pakketten.

Configuration Syntax

access-list *access-list-number* {permit | deny} *source* {source-mask}
ACL toepassen op een interface **ip access-group** *access-list-number* {in | out}

Voorbeeld van een standaard IP-toegangslijst



Configuratie:

In dit voorbeeld definiëren we een standaard toegangslijst die alleen netwerk 10.0.0.0/8 toegang geeft tot de server (op de Fa0/1-interface)

Definieer welke bron mag doorgaan:

Router(config)#**access-list 1 permit 10.0.0.0 0.255.255.255**

(er is altijd een impliciete weigering van al het andere verkeer aan het einde van elke ACL, zodat we verboden verkeer niet hoeven te definiëren)

Apply this ACL to an interface:

Router(config)#**interface Fa0/1**
Router(config-if)#**ip access-group 1 out**

De ACL 1 wordt toegepast om alleen pakketten van 10.0.0.0/8 uit de Fa0/1-interface te laten gaan terwijl al het andere verkeer wordt geweigerd. Dus kunnen we deze ACL toepassen op een andere interface, Fa0 / 2 bijvoorbeeld? Nou, we kunnen het maar moeten het niet doen omdat gebruikers toegang hebben tot de server vanuit een andere interface (s0-interface, bijvoorbeeld). We kunnen

dus begrijpen waarom een standaard toegangslijst dicht bij de bestemming moet worden toegepast.

Opmerking: De "0.255.255.255" is het jokertekenmaskergedeelte van netwerk "10.0.0.0". We zullen later leren hoe we het wildcardmasker kunnen gebruiken.

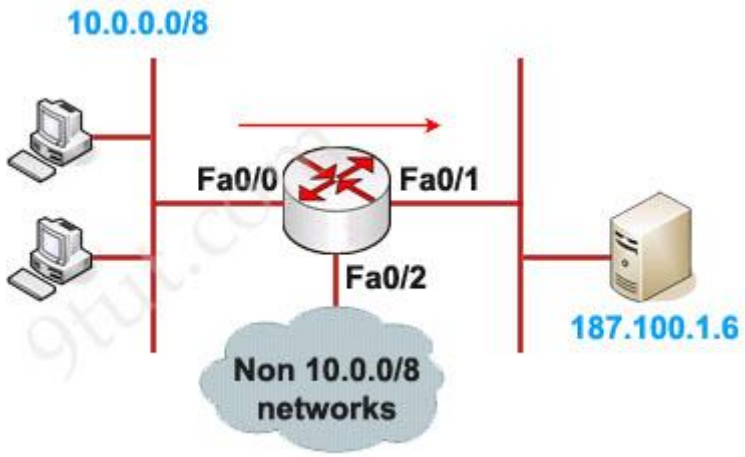
Uitgebreide IP-toegangslijst

Uitgebreide IP-lijsten (100-199) controleren zowel bron- als doeladressen, specifieke UDP/TCP/IP-protocollen en doelpoorten.

Configuration Syntax

access-list *access-list-number* {permit | deny} *protocol* source {source-mask} destination {destination-mask} [eq destination-port]

Example of Extended IP Access List



In dit voorbeeld maken we een uitgebreide ACL die FTP-verkeer van netwerk 10.0.0.0/8 weigert, maar ander verkeer doorlaat.

Opmerking: FTP gebruikt TCP op poort 20 &; 21.

Definieer welk protocol, bron, bestemming en poort worden geweigerd:

Router(config)#**access-list 101 deny tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 21**

Router(config)#**access-list 101 deny tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 20**

Router(config)#**access-list 101 permit ip any any**

Pas deze ACL toe op een interface:

Router(config)#**interface Fa0/1**
Router(config-if)#**ip access-group 101 out**

Merk op dat we expliciet ander verkeer moeten toestaan (access-list 101 allow ip any any) omdat er een "deny all" commando is aan het einde van elke ACL. Zoals we kunnen zien, is de bestemming van de bovenstaande toegangslijst "187.100.1.6 0.0.0.0" die een host specificeert. We kunnen in plaats daarvan "host 187.100.1.6" gebruiken. We zullen het wildcardmasker later bespreken. Samenvattend vindt u hieronder het bereik van de standaard- en uitgebreide toegangslijst **Access list type**	**Range**
Standard	1-99, 1300-1999

Extended	100-199, 2000-2699

Lijst met benoemde IP-toegangsadressen

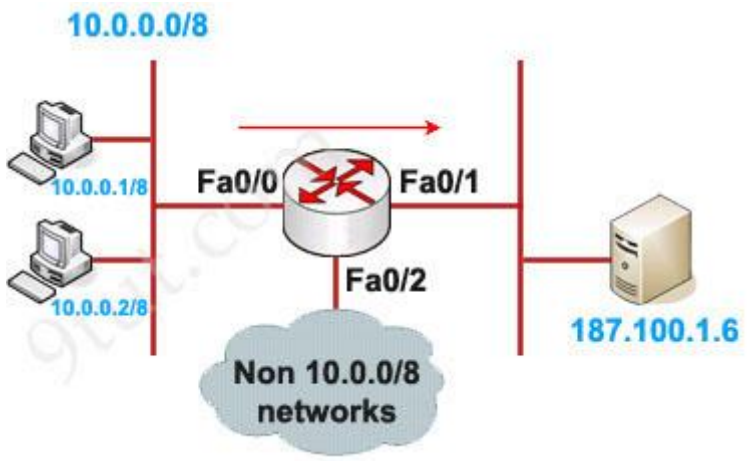
Hierdoor kunnen standaard en uitgebreide ACL's namen krijgen in plaats van nummers

Configuratiesyntaxis voor benoemde IP-toegangslijst

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

Voorbeeld van een lijst met benoemde IP-toegang

Dit is een voorbeeld van het gebruik van een benoemde ACL om al het verkeer te blokkeren, behalve de Telnet-verbinding van host 10.0.0.1/8 naar host 187.100.1.6.



Definieer de ACL:

Router(config)#**ip access-list extended in_to_out permit tcp host 10.0.0.1 host 187.100.1.6 eq telnet**

(merk op dat we 'telnet' kunnen gebruiken in plaats van poort 23)

Pas deze ACL toe op een interface:

Router(config)#**interface Fa0/0**

Router(config-if)#**ip access-group in_to_out in**

Waar plaats je een toegangslijst?

Standaard IP-toegangslijst moet dicht bij de bestemming worden geplaatst.

Uitgebreide IP-toegangslijsten moeten dicht bij de bron worden geplaatst.

Hoeveel toegangslijsten kunnen worden gebruikt?

U kunt één toegangslijst hebben per protocol, per richting en per interface. U kunt bijvoorbeeld geen twee toegangslijsten hebben in de binnenkomende richting van de Fa0/0-interface. U kunt echter één inkomende en één uitgaande toegangslijst toepassen op Fa0/0.

Hoe gebruik ik het jokertekenmasker?

Jokertekenmaskers worden gebruikt met toegangslijsten om een host, netwerk of deel van een netwerk op te geven.

De nullen en enen in een jokerteken bepalen of de overeenkomstige bits in het IP-adres moeten worden gecontroleerd of genegeerd voor ACL-doeleinden. We willen bijvoorbeeld een standaard ACL maken die alleen netwerk 172.23.16.0/20 doorlaat. We moeten een ACL schrijven, zoiets als dit:

access-list 1 permit 172.23.16.0 255.255.240.0

Natuurlijk kunnen we geen subnetmasker in een ACL schrijven, we moeten het omzetten in jokertekenmasker door alle bits 0 naar 1 en alle bits 1 naar 0 te converteren.

255 = 1111 1111 -> convert into 0000 0000

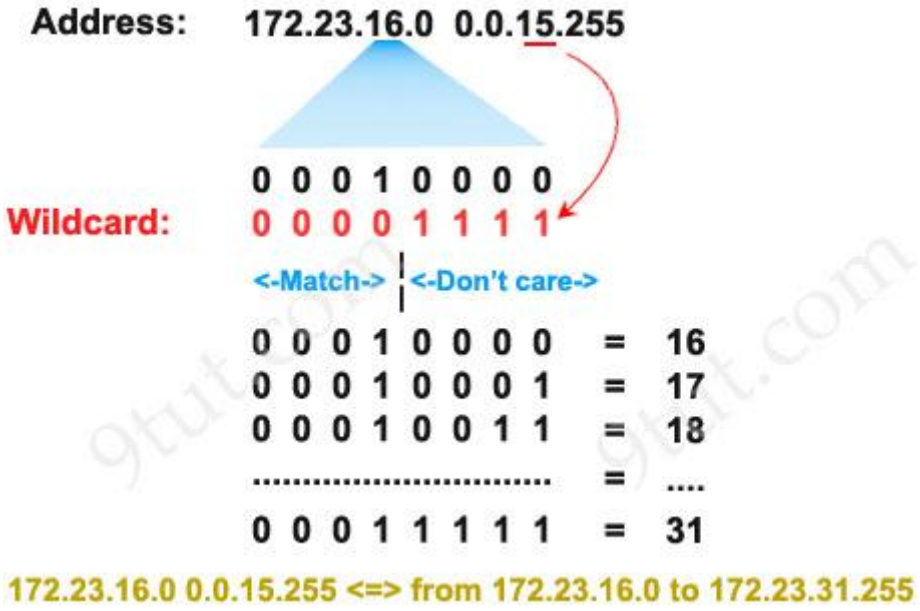
240 = 1111 0000 -> convert into 0000 1111

0 = 0000 0000 -> convert into 1111 1111

Daarom kan 255.255.240.0 in jokertekenmasker worden geschreven als 00000000.000000000.00001111.11111111 = 0.0.15.255

Vergeet niet dat voor het jokertekenmasker **1's I DON'T CARE zijn en 0's I CARE**. Laten we nu ons wildcardmasker analyseren.

Twee eerste octetten zijn allemaal 0's, wat betekent dat we om het netwerk **172 geven**.**23.x.x**. Het derde octet, 15 (0000 1111 in binair), betekent dat we geven om de eerste 4 bits, maar niet om de laatste 4 bits, dus we staan het derde octet toe in de vorm van 0001xxxx (minimum:00010000 = 16; maximum: 0001111 = 31).



Het vierde octet is 255 (alle 1 bits) dat betekent dat het me niet kan schelen.

Daarom **varieert netwerk 172.23.16.0 0.0.15.255 van 172.23.16.0** tot **172.23.31.255.**

Enkele aanvullende voorbeelden:

+ Blokkeer TCP-pakketten op poort 30 van elke bron naar elke bestemming:

Router(config)#access-list 101 deny tcp any any eq 30

+ Sta alle IP-pakketten in netwerk 192.23.130.128 met subnetmasker 255.255.255.248 toe aan elk netwerk:

Router(config)#access-list 101 permit ip 192.23.130.128 0.0.0.7 any

Pas de toegangsbeheerlijst toe op een interface:

Router(config)#interface fastEthernet0/0

Router(config-if)#ip access-group 101 in

Opmerking: Een ACL die op de hoofdinterface wordt toegepast, heeft geen invloed op het verkeer van subinterfaces. Als we verkeer op subinterfaces willen filteren, moeten we ACL aan elke subinterface afzonderlijk toewijzen.

Er zijn enkele verschillen tussen genummerde ACL en benoemde ACL:

+ Alleen genummerde ACL wordt ondersteund op VTY-regels (met behulp van de opdracht access-class)+ Alleen benoemde ACL ondersteunt niet-aaneengesloten poorten (hiermee kunt u niet-aaneengesloten poorten opgeven in één ACL-instructie). Bijvoorbeeld:Router(config)#ip access-list extended noncontiguousPortsRouter(config-ext-nacl)# allow tcp any eq telnet ftp any eq 23 45 34+ Alleen met named ACL kunnen we eenvoudig een individuele vermelding verwijderen. Bijvoorbeeld:

R1# show access-list

Standard IP access list nat_traffic
10 permit 10.1.0.0, wildcard bits 0.0.255.255
20 permit 10.2.0.0, wildcard bits 0.0.255.255
30 permit 10.3.0.0, wildcard bits 0.0.255.255

Om vervolgens de tweede instructie te verwijderen (de regel "20 permit 10.2.0.0, wildcard bits 0.0.255.255") hoeven we alleen maar "no 20" te typen:

R1(config)#ip access-list standard nat_traffic
R1(config-std-nacl)#no 20

Maar voor genummerde ACL moeten we de hele ACL opnieuw maken wanneer items worden verplaatst.