ACL (Erişim Kontrol Listeleri) Konfigürasyon Rehberi

Hazırlayan: Furkan Yaşar *LinkedIn*

Bu rehber, ACL kavramları ve IPv4 ACL konfigürasyonları için temel bilgileri içermektedir.

1. ACL'lerin Amacı

TEMEL KAVRAM: ACL (Erişim Kontrol Listesi), paket başlığındaki bilgilere dayanarak paketleri filtrelemek için kullanılan bir dizi IOS komutudur.

ACL Nedir?

ACL'ler, ağ güvenlik politikasının bir parçası olarak kullanılır ve şu özelliklere sahiptir:

- Varsayılan olarak, bir router'da hiçbir ACL yapılandırılmamıştır
- ACL bir arayüze uygulandığında, router paketleri değerlendirerek iletip iletmeyeceğine karar verir
- ACL'ler, permit (izin verme) veya deny (reddetme) ifadelerinden oluşan sıralı bir listedir (ACE'ler)
- Bu işleme paket filtreleme denir

ACL'lerin Kullanım Amaçları

Görev	Açıklama
Ağ Performansı	Ağ trafiğini sınırlandırarak performansı artırma
Trafik Akış Kontrolü	Trafik akışını kontrol etme
Temel Güvenlik	Ağ erişimi için temel güvenlik sağlama
Trafik Türüne Göre Filtreleme	Trafik türüne göre filtreleme yapma
Host Tarama	Host'ları tarayarak ağ servislerine erişimi kontrol etme
Önceliklendirme	Belirli trafik sınıflarına öncelik verme

Paket Filtreleme

Paket filtreleme, gelen ve/veya giden paketleri analiz ederek ve verilen kriterlere göre onları ileterek veya atarak bir ağa erişimi kontrol eder.

Cisco router'ları iki tür ACL destekler:

- **Standart ACL'ler** - Sadece kaynak IPv4 adresini kullanarak Layer 3'te filtreleme yapar
- **Genişletilmiş ACL'ler** - Kaynak ve/veya hedef IPv4 adresini kullanarak Layer 3'te filtreleme yapar. Ayrıca TCP, UDP portları ve isteğe bağlı protokol türü bilgilerini kullanarak Layer 4'te daha ince kontrol sağlar

OSI Modeli ve ACL'ler

Katman	İsim	ACL Türü
7	Uygulama	Genişletilmiş ACL'ler
6	Sunum	
5	Oturum	
4	Taşıma	Genişletilmiş ACL'ler (port tabanlı)
3	Ağ	Standart ve Genişletilmiş ACL'ler
2	Veri Bağlantısı	-
1	Fiziksel	-

ACL İşleyişi

ACL'ler, gelen arayüzlere giren paketler, router üzerinden iletilen paketler ve router'ın çıkış arayüzlerinden çıkan paketler için ek kontrol sağlayan kurallar kümesini tanımlar.

> **ÖNEMLİ:** ACL'ler router'ın kendisinden kaynaklanan paketler üzerinde etki etmez.

- **Gelen ACL** - Paketler çıkış arayüzüne yönlendirilmeden önce filtreler. Paket atılırsa yönlendirme arama yükünden tasarruf ettiği için verimlidir
- **Giden ACL** - Gelen arayüzden bağımsız olarak, yönlendirmeden sonra paketleri filtreler

ACL İşlem Adımları

Bir ACL bir arayüze uygulandığında şu işlem adımlarını izler:

1. Router, paket başlığından kaynak IPv4 adresini çıkarır
2. Router, ACL'nin en üstünden başlar ve kaynak IPv4 adresini her ACE ile sırayla karşılaştırır
3. Bir eşleşme olduğunda, router talimatı yerine getirir (pakete izin verir veya reddeder) ve ACL'de kalan ACE'ler (varsa) analiz edilmez
4. Kaynak IPv4 adresi ACL'deki herhangi bir ACE ile eşleşmezse, paket atılır çünkü tüm ACL'lere otomatik olarak uygulanan örtük bir reddetme ACE'si vardır

> **KRİTİK BİLGİ:** Bir ACL'nin son ACE ifadesi her zaman tüm trafiği engelleyen örtük bir deny'dır. Gizlidir ve yapılandırmada görüntülenmez. ACL'nin en az bir permit ifadesi olmalıdır, aksi takdirde örtük deny ACE ifadesi nedeniyle tüm trafik reddedilir.

2. ACL'lerde Wildcard Maskeleri

Wildcard Mask Genel Bakış

Wildcard mask, bir IPv4 adresindeki hangi bitlerin eşleştirileceğini belirlemek için ANDing işlemini kullanması bakımından bir alt ağ maskesine benzer. Ancak, alt ağ maskesinden farklı olarak:

- **Wildcard mask biti 0** - Adresteki karşılık gelen bit değeriyle eşleş
- **Wildcard mask biti 1** - Adresteki karşılık gelen bit değerini yoksay

Wildcard Mask Örnekleri

Wildcard Mask	Son Oktet (Binary)	Anlamı (0 - eşleş, 1 - yoksay)
0.0.0.0	00000000	Tüm oktetleri eşleştir
0.0.0.63	00111111	İlk üç okteti eşleştir, son oktetin en soldaki iki bitini eşleştir, son 6 biti yoksay
0.0.0.15	00001111	İlk üç okteti eşleştir, son oktetin dört sol bitini eşleştir, son 4 biti yoksay
0.0.0.248	11111000	İlk üç okteti eşleştir, son oktetin beş sol bitini yoksay, son 3 biti eşleştir
0.0.0.255	11111111	İlk üç okteti eşleştir, son okteti yoksay

Wildcard Mask Türleri

Bir Host'u Eşleştirmek İçin

```
access-list 10 permit 192.168.1.1 0.0.0.0 // Sadece 192.168.1.1 adresine izin ver
```

Veya kısa yazım:

```
access-list 10 permit host 192.168.1.1 // 'host' anahtar kelimesi ile tek bir adrese izin ver
```

Bir IPv4 Alt Ağını Eşleştirmek İçin

```
access-list 10 permit 192.168.1.0 0.0.0.255 // 192.168.1.0/24 ağındaki tüm adreslere izin ver
```

Bir IPv4 Adres Aralığını Eşleştirmek İçin

```
access-list 10 permit 192.168.16.0 0.0.15.255 // 192.168.16.0'dan 192.168.31.0'a kadar olan ağlara izin ver
```

Bu wildcard mask, 192.168.16.0'dan 192.168.31.0'a kadar olan tüm ağları kapsar (16 ağ).

Wildcard Mask Hesaplama

Wildcard maskları hesaplamak için kısayol yöntemi, alt ağ maskesini 255.255.255.255'tan çıkarmaktır.

Örnek 1: 192.168.3.0/24 ağı için wildcard mask

```
255.255.255.255
- 255.255.255.0
-----------
  0.0.0.255
```

Örnek 2: 192.168.3.32/28 ağı için wildcard mask

```
255.255.255.255
- 255.255.255.240
-----------
  0.0.0.15
```

Örnek 3: 192.168.10.0/23 ağı için wildcard mask (192.168.10.0 ve 192.168.11.0 ağlarını kapsar)

```
255.255.255.255
- 255.255.254.0
-----------
  0.0.1.255
```

Wildcard Mask Anahtar Kelimeleri

Cisco IOS, wildcard masking'in en yaygın kullanımlarını belirlemek için iki anahtar kelime sağlar:

- **host** - 0.0.0.0 maskesinin yerine kullanılır. Sadece bir host adresini filtrelemek için tüm IPv4 adres bitlerinin eşleşmesi gerektiğini belirtir
- **any** - 255.255.255.255 maskesinin yerine kullanılır. Tüm IPv4 adresini yoksaymayı veya herhangi bir adresi kabul etmeyi belirtir

3. ACL Oluşturma Kuralları

Arayüz Başına Sınırlı Sayıda ACL

Bir router arayüzüne uygulanabilen ACL sayısında sınır vardır. Örneğin, çift yığınlı (IPv4 ve IPv6) bir router arayüzü, şekilde gösterildiği gibi en fazla dört ACL uygulanabilir:

- Bir giden IPv4 ACL
- Bir gelen IPv4 ACL
- Bir gelen IPv6 ACL
- Bir giden IPv6 ACL

> **NOT:** ACL'ler her iki yönde de yapılandırılmak zorunda değildir. Arayüze uygulanan ACL'ersin sayısı ve yönü, kuruluşun güvenlik politikasına bağlı olacaktır.

ACL En İyi Uygulamaları

Kural	Fayda
ACL'leri kurumsal güvenlik politikalarına dayandırın	Kurumsal güvenlik yönergelerini uyguladığınızdan emin olur
ACL'nin ne yapmasını istediğinizi yazın	Yanlışlıkla potansiyel erişim sorunları oluşturmaktan kaçınmanıza yardımcı olur
Tüm ACL'lerinizi oluşturmak, düzenlemek ve kaydetmek için bir metin düzenleyici kullanın	Yeniden kullanılabilir ACL'lerden oluşan bir kitaplık oluşturmanıza yardımcı olur
ACL'leri **remark** (açıklama) komutunu kullanarak belgeleyin	Bir ACE'nin amacını anlamanıza (ve başkalarının anlamasına) yardımcı olur
ACL'leri üretim ağında uygulamadan önce bir geliştirme ağında test edin	Maliyetli hatalardan kaçınmanıza yardımcı olur

4. IPv4 ACL Türleri

Standart ve Genişletilmiş ACL'ler

İtwo tür IPv4 ACL vardır:

- **Standart ACL'ler** - Sadece kaynak IPv4 adresine dayanarak paketlere izin verir veya reddeder
- **Genişletilmiş ACL'ler** - Kaynak IPv4 adresi ve hedef IPv4 adresi, protokol türü, kaynak ve hedef TCP veya UDP portları ve daha fazlasına dayanarak paketlere izin verir veya reddeder

Numaralı ve İsimli ACL'ler

Numaralı ACL'ler:

- 1-99 veya 1300-1999 arası numaralandırılmış ACL'ler standart ACL'lerdir
- 100-199 veya 2000-2699 arası numaralandırılmış ACL'ler genişletilmiş ACL'lerdir

İsimli ACL'ler:

- ACL'leri yapılandırırken tercih edilen yöntemdir
- Standart ve genişletilmiş ACL'ler, ACL'nin amacı hakkında bilgi sağlamak için isimlendirilebilir
- Örneğin, genişletilmiş bir ACL'yi FTP-FILTER olarak adlandırmak, 100 numaralı bir ACL'ye sahip olmaktan çok daha iyidir

```
R1(config)# ip access-list extended FTP-FILTER // FTP filtreleme için genişletilmiş ACL oluştur
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp // FTP trafiğine izin ver
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data // FTP veri trafiğine izin ver
```

ACL'lerin Yerleştirilmesi

- Her ACL, verimlilik üzerinde en büyük etkiye sahip olduğu yere yerleştirilmelidir
- Genişletilmiş ACL'ler, filtrelenecek trafiğin kaynağına mümkün olduğunca yakın konumlandırılmalıdır
- Standart ACL'ler hedefe mümkün olduğunca yakın konumlandırılmalıdır

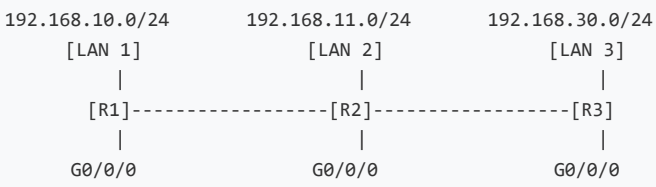
ACL Yerleşimini Etkileyen Faktörler

Faktör	Açıklama
Kurumsal kontrolün kapsamı	ACL'nin yerleşimi, kuruluşun hem kaynak hem de hedef ağlar üzerinde kontrolü olup olmamasına bağlı olabilir
İlgili ağların bant genişliği	Bant genişliği tüketen trafiğin iletimini önlemek için istenmeyen trafiği kaynağında filtrelemek istenebilir
Yapılandırma kolaylığı	ACL'yi hedefte uygulamak daha kolay olabilir, ancak trafik bant genişliğini gereksiz yere kullanır. Trafiğin kaynaklandığı her router'da genişletilmiş bir ACL kullanılabilir. Bu, trafiği kaynağında filtreleyerek bant genişliğinden tasarruf sağlar, ancak birden çok router'da genişletilmiş ACL'ler oluşturulmasını gerektirir

Standart ACL Yerleştirme Örneği

192.168.10.0/24 ağından kaynaklanan trafiğin 192.168.30.0/24 ağına ulaşmasını engellemek isteyen bir yönetici, standart bir ACL'yi R3 router'ına yerleştirecektir.

Örnek Ağ Topolojisi: Standart ACL Yerleşimi

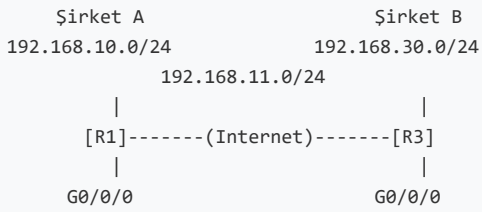


Bu senaryoda, R3'ün G0/0/0 arayüzüne (hedefe yakın) uygulanan bir standart ACL, 192.168.10.0/24 ağından gelen trafiği engelleyecektir.

Genişletilmiş ACL Yerleştirme Örneği

Şirket A, 192.168.11.0/24 ağından Şirket B'nin 192.168.30.0/24 ağına Telnet ve FTP trafiğini reddederken diğer tüm trafiğe izin vermek istemektedir. Genişletilmiş bir ACL R1 üzerine yerleştirilebilir.

Örnek Ağ Topolojisi: Genişletilmiş ACL Yerleşimi



Bu senaryoda, R1'in G0/0/1 arayüzüne (kaynağa yakın) uygulanan bir genişletilmiş ACL, Telnet ve FTP trafiğini kaynağında engelleyecektir.

5. Standart IPv4 ACL'leri Yapılandırma

ACL Oluşturma

Tüm erişim kontrol listeleri (ACL'ler) planlanmalıdır. Karmaşık bir ACL yapılandırırken şunlar önerilir:

1. Bir metin düzenleyici kullanın ve uygulanacak politikanın ayrıntılarını yazın
2. Bu görevleri yerine getirmek için IOS yapılandırma komutlarını ekleyin
3. ACL'yi belgelemek için açıklamalar ekleyin
4. Komutları cihaza kopyalayıp yapıştırın
5. Bir ACL'yi her zaman istenen politikayı doğru şekilde uyguladığından emin olmak için iyice test edin

Numaralı Standart IPv4 ACL Sözdizimi

```
Router(config)# access-list access-list-number {deny | permit | remark text} source [source-wildcard] [log]
```

Parametre	Açıklama
access-list-number	Numara aralığı 1-99 veya 1300-1999
deny	Koşul eşleşirse erişimi reddeder
permit	Koşul eşleşirse erişime izin verır
remark text	(İsteğe bağlı) Belgeme amaçlı metin girişi
source	Filtrelenecek kaynak ağ veya host adresini tanımlar
source-wildcard	(İsteğe bağlı) Kaynağa uygulanan 32-bit wildcard maskesi
log	(İsteğe bağlı) ACE eşleştiğinde bilgilendirici mesaj oluşturur ve gönderir

> **NOT:** Numaralı bir standart ACL'yi kaldırmak için **no access-list access-list-number** global yapılandırma komutunu kullanın.

İsimli Standart IPv4 ACL Sözdizimi

İsimli bir standart ACL oluşturmak için **ip access-list standard** komutunu kullanın:

- ACL isimleri alfasayısal, büyük/küçük harfe duyarlı ve benzersiz olmalıdır
- ACL isimlerini büyük harfle yazmak gerekli değildir ancak running-config çıktısını görüntülerken öne çıkmalarını sağlar

```
Router(config)# ip access-list standard access-list-name // İsimli standart ACL oluşturur
```

Standart IPv4 ACL Uygulama

Bir standart IPv4 ACL yapılandırıldıktan sonra, bir arayüze veya özelliğe bağlanmalıdır.

- Numaralı veya isimli bir standart IPv4 ACL'yi bir arayüze bağlamak için **ip access-group** komutu kullanılır
- Bir ACL'yi bir arayüzden kaldırmak için önce **no ip access-group** arayüz yapılandırma komutunu girin

```
Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out} // ACL'yi arayüze uygular
```

Numaralı Standart ACL Örneği

Aşağıdaki örnek ACL, 192.168.10.10 host'undan ve 192.168.20.0/24 ağındaki tüm host'lardan gelen trafiğe R1 router'ının serial 0/1/0 çıkış arayüzünden izin verir.

```
R1(config)# access-list 10 remark ACE permits ONLY host 192.168.10.10 to the internet // Açıklama ekler
R1(config)# access-list 10 permit host 192.168.10.10 // Belirli bir host'a izin verir
R1(config)# access-list 10 remark ACE permits all host in LAN 2 // Açıklama ekler
R1(config)# access-list 10 permit 192.168.20.0 0.0.0.255 // Tüm bir ağa izin verir
R1(config)# interface Serial 0/1/0 // Seri arayüze geçer
R1(config-if)# ip access-group 10 out // ACL'yi çıkış trafiğine uygular
```

İsimli Standart ACL Örneği

Aynı filtreleme işlemini isimli ACL ile gerçekleştirme:

```
R1(config)# ip access-list standard PERMIT-ACCESS // İsimli standart ACL oluşturur
R1(config-std-nacl)# remark ACE permits host 192.168.10.10 // Açıklama ekler
R1(config-std-nacl)# permit host 192.168.10.10 // Belirli bir host'a izin verir
R1(config-std-nacl)# remark ACE permits all hosts in LAN 2 // Açıklama ekler
R1(config-std-nacl)# permit 192.168.20.0 0.0.0.255 // Tüm bir ağa izin verir
R1(config)# interface Serial 0/1/0 // Seri arayüze geçer
R1(config-if)# ip access-group PERMIT-ACCESS out // İsimli ACL'yi çıkış trafiğine uygular
```

6. IPv4 ACL'leri Değiştirme

Bir ACL'yi Değiştirmek İçin İki Yöntem

Bir ACL yapılandırıldıktan sonra değiştirilmesi gerekebilir. Birden çok ACE'ye sahip ACL'ler yapılandırılması karmaşık olabilir. Bazen yapılandırılan ACE beklenen davranışları sağlamaz.

Bir ACL'yi değiştirirken kullanılacak iki yöntem vardır:

1. Bir metin düzenleyici kullanın
2. Sıra numaralarını kullanın

Metin Düzenleyici Yöntemi

Birden çok ACE'ye sahip ACL'ler bir metin düzenleyicide oluşturulmalıdır. Bu, gerekli ACE'leri planlamanıza, ACL'yi oluşturmanıza ve ardından router arayüzüne yapıştırmanıza olanak tanır. Ayrıca bir ACL'yi düzenleme ve düzeltme görevlerini basitleştirir.

Sıra Numarası Yöntemi

ACL ACE, ACL sıra numaraları kullanılarak silinebilir veya eklenebilir.

- Bir ACL'yi düzenlemek için **ip access-list standard** komutunu kullanın
- İfadeler mevcut bir sıra numarası kullanılarak üzerine yazılamaz. Mevcut ifade önce **no [sequence-number]** komutuyla silinmelidir. Ardından doğru ACE sıra numarası kullanılarak eklenebilir

İsimli ACL'yi Değiştirme Örneği

İsimli ACL'ler de ACE'leri silmek ve eklemek için sıra numaralarını kullanabilir. Örnekte 192.168.10.5 host'unu reddetmek için bir ACE eklenmiştir.

```
R1# show access-lists // ACL'leri ve sıra numaralarını gösterir
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255

R1# configure terminal // Yapılandırma moduna geçer
R1(config)# ip access-list standard NO-ACCESS // İsimli ACL'yi düzenlemek için girer
R1(config-std-nacl)# 15 deny 192.168.10.5 // 15 numaralı yeni bir ACE ekler
R1(config-std-nacl)# end // Yapılandırma modundan çıkar

R1# show access-lists // Güncellenmiş ACL'yi gösterir
Standard IP access list NO-ACCESS
    15 deny  192.168.10.5
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
```

ACL İstatistikleri

show access-lists komutu, eşleşen her ifade için istatistikleri gösterir.

- Örtük deny any ifadesi herhangi bir istatistik göstermez. Kaç örtük reddedilmiş paketin eşleştiğini izlemek için manuel olarak **deny any** komutunu yapılandırmanız gerekir
- ACL istatistiklerini temizlemek için **clear access-list counters** komutunu kullanın

```
R1# show access-lists // ACL istatistiklerini gösterir
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10 (20 matches)
    20 permit 192.168.10.0, wildcard bits 0.0.0.255 (64 matches)

R1# clear access-list counters NO-ACCESS // ACL istatistiklerini sıfırlar

R1# show access-lists // Sıfırlanmış istatistikleri gösterir
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
```

7. Standart IPv4 ACL ile VTY Portlarının Güvenliği

access-class Komutu

Standart bir ACL, aşağıdaki iki adımı uygulayarak bir cihaza uzaktan yönetim erişimini güvenli hale getirebilir:

1. Hangi yönetim host'larının uzaktan erişime izin verilmesi gerektiğini belirlemek için bir ACL oluşturun
2. ACL'yi vty hatlarındaki gelen trafiğe uygulayın

```
R1(config-line)# access-class {access-list-number | access-list-name} { in | out } // VTY erişimini kontrol eder
```

VTY Erişim Güvenliği Örneği

Bu örnek, vty trafiğini filtrelemek için bir ACL'nin nasıl yapılandırılacağını gösterir.

```
R1(config)# username ADMIN secret class // Yerel kullanıcı oluşturur
R1(config)# ip access-list standard ADMIN-HOST // İsimli standart ACL oluşturur
R1(config-std-nacl)# remark This ACL secures incoming vty lines // Açıklama ekler
R1(config-std-nacl)# permit 192.168.10.10 // Belirli bir host'a izin verir
R1(config-std-nacl)# deny any // Diğer tüm host'ları reddeder
R1(config)# line vty 0 4 // VTY hatlarına geçer
R1(config-line)# login local // Yerel kimlik doğrulama kullanır
R1(config-line)# transport input telnet // Telnet erişimine izin verir
R1(config-line)# access-class ADMIN-HOST in // ACL'yi VTY girişine uygular
```

VTY Port Güvenliğini Doğrulama

Vty hatlarına erişimi kısıtlayan bir ACL yapılandırıldıktan sonra, beklendiği gibi çalıştığını doğrulamak önemlidir.

ACL istatistiklerini doğrulamak için **show access-lists** komutunu verin.

```
R1# show access-lists // ACL istatistiklerini gösterir
Standard IP access list ADMIN-HOST
    10 permit 192.168.10.10 (2 matches)
    20 deny any (2 matches)
```

8. Genişletilmiş IPv4 ACL'leri Yapılandırma

Genişletilmiş ACL'ler

Genişletilmiş ACL'ler daha fazla kontrol sağlar. Kaynak adresi, hedef adresi, protokol (IP, TCP, UDP, ICMP) ve port numarasına göre filtreleme yapabilirler.

Genişletilmiş ACL'ler şu şekilde oluşturulabilir:

- **Numaralı Genişletilmiş ACL** - **access-list access-list-number** global yapılandırma komutu kullanılarak oluşturulur
- **İsimli Genişletilmiş ACL** - **ip access-list extended access-list-name** komutu kullanılarak oluşturulur

Protokoller ve Portlar

Genişletilmiş ACL'ler internet protokolleri ve portları üzerinde filtreleme yapabilir. Karmaşık bir ACE girerken yardım almak için **?** kullanın.

Protokol ve Port Numarası Yapılandırma Örnekleri

Genişletilmiş ACL'ler farklı port numarası ve port adı seçeneklerinde filtreleme yapabilir.

Bu örnek, HTTP trafiğini filtrelemek için 100 numaralı genişletilmiş bir ACL yapılandırır. İlk ACE **www** port adını kullanır. İkinci ACE **80** port numarasını kullanır. Her iki ACE de tam olarak aynı sonucu elde eder.

```
R1(config)# access-list 100 permit tcp any any eq www // HTTP trafiğine port adıyla izin verir
R1(config)# access-list 100 permit tcp any any eq 80 // HTTP trafiğine port numarasıyla izin verir
```

SSH (port numarası 22) veya HTTPS (port numarası 443) gibi belirli bir protokol adı listelenmediğinde port numarasını yapılandırmak gerekir:

```
R1(config)# access-list 100 permit tcp any any eq 22 // SSH trafiğine izin verir
R1(config)# access-list 100 permit tcp any any eq 443 // HTTPS trafiğine izin verir
```

Numaralı Genişletilmiş IPv4 ACL Uygulama

Bu örnekte, ACL 192.168.10.0 ağından herhangi bir hedefe hem HTTP hem de HTTPS trafiğine izin verir.

Genişletilmiş ACL'ler çeşitli konumlara uygulanabilir. Ancak, genellikle kaynağa yakın uygulanırlar. Burada ACL 110, R1 G0/0/0 arayüzüne gelen olarak uygulanır.

```
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq www // HTTP trafiğine izin verir
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq 443 // HTTPS trafiğine izin verir
R1(config)# interface g0/0/0 // Arayüze geçer
R1(config-if)# ip access-group 110 in // ACL'yi gelen trafiğe uygular
```

TCP Established Genişletilmiş ACL

TCP, **established** (yerleşik) anahtar kelimesini kullanarak temel stateful güvenlik duvarı hizmetleri de gerçekleştirebilir.

- **established** anahtar kelimesi, iç trafiğin iç özel ağdan çıkmasını ve dönen yanıt trafiğinin iç özel ağa girmesini sağlar
- Dış host tarafından oluşturulan ve iç host ile iletişim kurmaya çalışan TCP trafiği reddedilir

TCP Established ACL Örneği

ACL 120, yalnızca iç host'lara dönen web trafiğine izin vermek için yapılandırılır. ACL daha sonra R1 G0/0/0 arayüzüne giden olarak uygulanır.

```
R1(config)# access-list 120 permit tcp any 192.168.10.0 0.0.0.255 established // Yerleşik TCP bağlantılarına izin verir
R1(config)# interface g0/0/0 // Arayüze geçer
R1(config-if)# ip access-group 120 out // ACL'yi giden trafiğe uygular
```

> **NOT:** Dönen TCP segmentinde ACK veya sıfırlama (RST) bayrak bitleri ayarlanmışsa, paketin mevcut bir bağlantıya ait olduğunu gösterir ve bir eşleşme oluşur.

İsimli Genişletilmiş IPv4 ACL Sözdizimi

Bir ACL'yi isimlendirmek, işlevini anlamayı kolaylaştırır. İsimli bir genişletilmiş ACL oluşturmak için **ip access-list extended** yapılandırma komutunu kullanın.

```
Router(config)# ip access-list extended access-list-name // İsimli genişletilmiş ACL oluşturur
```

İsimli Genişletilmiş IPv4 ACL Örneği

Aşağıdaki topoloji, iki isimli genişletilmiş IPv4 ACL'nin bir arayüze yapılandırılmasını ve uygulanmasını göstermek için kullanılır:

- **SURFING** - İç HTTP ve HTTPS trafiğinin internete çıkmasına izin verir
- **BROWSING** - Yalnızca iç host'lara dönen web trafiğine izin verirken, R1 G0/0/0 arayüzünden çıkan diğer tüm trafik örtük olarak reddedilir

```
R1(config)# ip access-list extended SURFING // İnternet gezintisi için ACL oluşturur
R1(config-ext-nacl)# Remark Permits inside HTTP and HTTPS traffic // Açıklama ekler
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80 // HTTP trafiğine izin verir
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443 // HTTPS trafiğine izin verir
R1(config-ext-nacl)# exit // Çıkış yapar
R1(config)#
R1(config)# ip access-list extended BROWSING // Tarama için ACL oluşturur
R1(config-ext-nacl)# Remark Only permit returning HTTP and HTTPS traffic // Açıklama ekler
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established // Yerleşik bağlantılara izin verir
R1(config-ext-nacl)# exit // Çıkış yapar
R1(config)# interface g0/0/0 // Arayüze geçer
R1(config-if)# ip access-group SURFING in // Gelen trafiğe SURFING ACL'sini uygular
R1(config-if)# ip access-group BROWSING out // Giden trafiğe BROWSING ACL'sini uygular
R1(config-if)# end // Yapılandırmadan çıkar
```

Genişletilmiş ACL'leri Doğrulama

Arayüzdeki ACL'yi ve uygulandığı yönü doğrulamak için **show ip interface** komutu kullanılır.

show access-lists komutu, ACL'lerin beklendiği gibi çalıştığını doğrulamak için kullanılabilir. Komut, bir ACE eşleştiğinde artan istatistik sayaçlarını görüntüler.

> **NOT:** ACL'nin işleyişini doğrulamak için trafik oluşturulmalıdır.

show running-config komutu, neyin yapılandırıldığını doğrulamak için kullanılabilir. Komut ayrıca yapılandırılmış açıklamaları da görüntüler.

Tüm ACL Komutları

Komut	Açıklama
access-list [numara] [permit/deny] [kaynak]	Numaralı ACL oluşturur
access-list [numara] remark [açıklama]	Numaralı ACL için açıklama ekler
access-list [numara] [permit/deny] [protokol] [kaynak] [hedef] [operatör] [port]	Genişletilmiş numaralı ACL kuralı ekler
ip access-list standard [isim]	İsimli standart ACL oluşturur
ip access-list extended [isim]	İsimli genişletilmiş ACL oluşturur
ip access-group [numara/isim] [in/out]	ACL'yi arayüze uygular
access-class [numara/isim] [in/out]	ACL'yi vty hatlarına uygular
show access-lists	ACL'leri ve istatistiklerini gösterir
show ip interface [arayüz]	Arayüz ACL yapılandırmasını gösterir
clear access-list counters [numara/isim]	ACL istatistiklerini temizler
remark [açıklama]	İsimli ACL için açıklama ekler
show running-config | include access-list	Yapılandırılmış ACL'leri gösterir

> **SON UYARI:** ACL'leri yapılandırırken daima test edin ve önce geliştirme ortamında uygulayın. Yanlış yapılandırılmış bir ACL ağ trafiğini kesintiye uğratabilir. ACL'lerde her zaman örtük reddetmeyi unutmayın ve en az bir izin ver ifadesi ekleyin.