

Cisco IOS ve PIX Firewall ACL Konfigürasyonu

TANER KOÇ-MEHMET ALİ KARAGÖL / İTÜ'BİDB 2010

IOS ACL Konfigürasyonu

ACL (Access Control List), network ataklarını engellemek ve network trafiğini kontrol etmek için kullanılan bir yapıdır. ACL'ler sayesinde networkümüze giren ve networkümüzden çıkan trafiği adres ve port bazlı olarak filtreleyebiliriz. Bu işlemi yapmak için kullanılan iki tür ACL vardır. Bunlar Standard ACL ve Extended ACL'dir.

Standard ACL

Standard ACL'ler 1-99 veya 1300-1999 arası numaralandırılan ve trafiği kontrol etmek için IP başlıklarındaki kaynak IP bilgisini inceleyen ACL'lerdir. Standard ACL'ler filtreleme işlemi sadece Layer 3 bilgisine bakarak gerçekleştirirler. Standard ACL'ler şu şekilde oluşturulur:

```
Router(config)# access-list {1-99} {permit/deny} source-addr [source-wildcard]
```

İlk bölüm, ACL numarasını ifade eder. İkinci bölüm, belirtilen kaynak IP adresine izin verilip verilmeyeceğini belirtir. Üçüncü bölüm, üzerinde işlem yapılacak kaynak IP adresini gösterir. Dördüncü kısımdaki wildcard maskesi ise üzerinde işlem yapılacak ip adres aralığını belirler.

```
AGCIYIZ_ACL(config)# ip access-list standard 70
```

```
AGCIYIZ_ACL(config-std-nacl)# permit 192.168.1.0 0.0.0.255
```

Extended ACL

Extended ACL'ler paketlerin Layer 3 ve Layer 4 bilgilerine göre filtreleme işlemi yaparlar. TCP veya UDP port bilgisini, kaynak ve hedef IP bilgisini içerirler. Extended ACL'ler şu şekilde düzenlenirler:

```
Router(config)# access-list {100-199} {permit/deny} protocol source-addr [source-wildcard] destination-addr [destination-wildcard] [operator operand]
```

İlk bölüm, extended ACL numarasını belirtir. İkinci bölüm, daha sonraki bölümlerde özellikleri belirtilen pakete izin verilip verilmeyeceğini gösterir. Üçüncü bölümde protokol türü ifade edilir. Protokol türü tcp, udp veya ip biçiminde belirtilebilir. Dördüncü bölümde filtre edilecek paketlerin kaynak IP adresleri ve IP aralığını belirten wildcard maskesi bulunur. Ardından hedef IP adresi ve aralığı belirtilir. Son olarak üzerinde filtreleme yapılacak port belirtilir.

```
AGCIYIZ_ACL(config)# ip access-list extended 125
```

```
AGCIYIZ_ACL(config-ext-nacl)#deny udp host 192.168.2.3 any
```

Named ACL

Standard ve extended ACL'leri isimle oluşturmamız mümkündür. Bunun için yapılacak konfigürasyon şöyledir:

```
Router(config)# ip access-list {standard/extended} name -of-ACL
```

Böylece alt konfigürasyon moduna girilir. Standard ACL konfigüre ediliyorsa;

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

Eğer extended ACL konfigüre ediliyorsa;

```
Router(config-ext-nacl)# {permit|deny} protocol source-addr [source-wildcard] destination-addr  
[destination-wildcard] [operator operand]
```

İsime oluşturulmuş ACL'lerin avantajı, yöneticinin bu sayede ACL alt konfigürasyonuna girerek spesifik bir girdiyi silebilmesidir. Numaralı ACL'lerde no komutuyla tüm ACL silinir. İsimle oluşturulmuş ACL'lerde istenilen bölüme yeni bir girdi eklenebilir. Oysa numaralı ACL'lerde yeni girdiler en sona eklenir.

```
Router(config-if)# ip access-group <access-list-name> {in|out}
```

Yukarıda Standard ACL'de 192.168.1.0 network'ündeki tüm IP'lere erişim izni veriliyor. Extended ACL'de 192.168.2.3 IP adresine sahip hosttan herhangi bir yere giden udp paketlerinin erişimi engelleniyor. Ancak şöyle bir şey var; IOS'larda access control list yazıldığında bu ACL'nin geçerli olması için herhangi bir interface altına, telnet erişimi için vty'ye, ip http server altına vs. uygulanması gerekmektedir. Aksi takdirde ne kadar access-list satırı yazmış olursak olalım yazdığımız satırların hiçbir önemi olmayacaktır. Örnek olarak;

```
AGCIYIZ_ACL# conf t  
AGCIYIZ_ACL(config)# ip access-list standard 70  
AGCIYIZ_ACL(config-std-nacl)# permit 192.168.1.0 0.0.0.255
```

Interface'ler için;

```
Router(config-if)# ip access-group <access-list-number> {in|out}
```

```
AGCIYIZ_ACL(config)# interface fastEthernet 0/3  
AGCIYIZ_ACL(config-if)# ip access-group 70 in
```

Telnet için;

```
Router(config-if)# access-class <access-list-number> {in|out}
```

```
AGCIYIZ_ACL(config)# line vty 0 4  
AGCIYIZ_ACL(config-line)# access-class 70 in
```

Sadece Standard ve Extended ACL'ler için örnek verdik ama Named ACL'ler de interface, telnet vs. altına uygulanabilir.

Aslında IOS'larda access-list yazarken önce access-list numarasını belirtip daha sonra da erişim kuralını yazmak zorunda olduğumuzu söylemiştik, ancak direkt olarak access-list numarasını belirttikten sonra da erişim kuralını aynı satıra yazabiliriz. Örnek olarak;

```
AGCIYIZ_ACL(config)# access-list 71 permit host 192.168.2.3
```

Standard ya da Extended olduğunu belirtmemize gerek yok çünkü zaten access-list numarasını yazdıktan sonra Standard mı Extended mi olduğunu anlayıp sonraki komutlar ona göre oluşturuluyor.

Ancak Named ACL yazmak için mutlaka ilk başta anlattığımız gibi önce ACL'yi oluşturup, bir bakıma ACL'nin içine girip, daha sonra erişim kuralını yazıyoruz. Örnek olarak;

```
AGCIYIZ_ACL(config)# ip access-list extended TCP_TRAFIGI  
AGCIYIZ_ACL(config-ext-nacl)# deny tcp any host 192.168.4.7 eq 3123
```

Not: ACL'lerin altında kapalı olarak, belirtilen özelliklerin dışında paketlerin drop edilmesini sağlayan kapalı bir komut olan implicit deny bulunur.

ASA, PIX veya 6500 ve 7600 serisi cihazlara modül olarak takılan FWSM'de access-list yazmak mantık olarak aynıdır fakat yazılış biçimi olarak değişiktir. IOS'larda önce ACL'yi oluşturup daha sonra aktif hale gelmesi için interface, telnet vs. altına uygulanması gerektiğin söylemiştik. Firewall'da bu durum biraz değişiktir.

ASA/PIX veya Blade Firewall için ACL Konfigürasyonu

PIX firewall temel konfigürasyonunu <http://www.agciyiz.net/index.php/genel/cisco-pix-firewall-temel-konfigurasyonu/> bağlantısındaki dokümanda anlatmıştık. O dokümandan yola çıkarak access-list konfigürasyonunu anlatacağız. Firewall'un iç ve dış ayağını belirttik, bununla beraber kendi networkümüzü ve dış networkü tanıtmış olduk. Şimdi iş sadece access-list'leri yazmaya ve bunu önceden tanımladığımız interface'lere uygulamaya geldi. Örnek olarak aşağıdaki access-list satırlarını yazmış olduğumuzu düşünürsek;

```
AGCIYIZ_FIREWALL(config)# access-list 135 extended permit ip host 193.194.135.150 host  
191.56.87.45  
AGCIYIZ_FIREWALL(config)# access-list 135 extended deny udp any any range snmp snmptrap  
AGCIYIZ_FIREWALL(config)# access-list 135 extended permit udp host 130.237.200.23 any range  
52000 52999  
AGCIYIZ_FIREWALL(config)# access-list 135 extended permit tcp host 194.27.40.44 host  
122.16.75.2.222 eq 1433  
AGCIYIZ_FIREWALL(config)# access-list 137 extended permit ip host 122.16.9.98 host  
193.194.135.153  
AGCIYIZ_FIREWALL(config)# access-list 137 extended permit tcp 122.16.85.144 255.255.255.240 any  
eq 8080  
AGCIYIZ_FIREWALL(config)# access-list 137 extended deny udp any any eq 135
```

Yazılan bu ACL satırları firewall'un ister iç ayağına ister dış ayağına uygulanabilir. Örneğin, 137 numaralı access-list içeriden dışarıya trafiği, 135 numaralı access-list de dışarıdan içeriye trafiği kontrol etmek için kullanılmak istenirse;

```
AGCIYIZ_FIREWALL(config)# access-group 135 in interface outside  
AGCIYIZ_FIREWALL(config)# access-group 137 in interface inside
```

satırlarının yazılması yeterli olacaktır.