IP 액세스 목록 구성 및 필터링

목차

소개

이 문서에서는 다양한 유형의 IP ACL(Access Control List)과 네트워크 트래픽을 필터링하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다. 언급한 개념은 Cisco IOS® 소프트웨어 릴리스 8.3 이상에 적용됩니다. 이는 각 액세스 목록 기능 아래에 표시됩니다.

사용되는 구성 요소

이 문서에서는 다양한 ACL 유형을 다룹니다. 그중에는 Cisco IOS 소프트웨어 릴리스 8.3부터 제공한 것도 있고, 이후에 출시된 소프트웨어 릴리스에서 도입한 것도 있습니다. 이는 각 유형에 관한 설명에 표시됩니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 규칙을 참조하십시오.

배경 정보

이 문서에서는 IP ACL(Access Control List)로 네트워크 트래픽을 필터링하는 방법을 설명합니다. IP ACL 유형에 관한 간단한 설명, 기능 제공 정보, 네트워크의 사용 예도 수록되어 있습니다.

> **참고:** RFC 1700에는 잘 알려진 포트의 할당된 번호가 포함되어 있습니다.RFC 1918에는 사설 인터넷의 주소 할당, 일반적으로 인터넷에서 볼 수 없는 IP 주소가 포함되어 있습니다.

> **참고:** 등록된 Cisco 사용자만 내부 정보에 액세스할 수 있습니다.

> **참고:** ACL을 사용하여 NAT(Network Address Translate)에 대한 트래픽을 정의하고, AppleTalk 또는 IPX와 같은 비 IP 프로토콜을 암호화하거나 필터링할 수도 있습니다. 이러한 기능에 대한 설명은 이 문서의 범위에 속하지 않습니다.

ACL 개념

마스크

마스크는 IP ACL의 IP 주소와 함께 사용되어 허용 및 거부해야 하는 항목을 지정합니다. 인터페이

스에서 IP 주소를 구성하기 위한 마스크는 255로 시작하고 왼쪽에 큰 값을 가집니다. 예를 들어, IP 주소는 255.255.255.224 마스크와 함께 10.165.202.129입니다. IP ACL에 대한 마스크는 그 반대입니다(예: mask 0.0.0.255). 이를 반전 마스크 또는 와일드카드 마스크라고도 합니다. 마스크 값을 이진(0s 및 1s)로 나누면 트래픽 처리 시 고려할 주소 비트가 결과에 따라 결정됩니다. 0은 주소 비트를 고려해야 함을 의미합니다(정확한 일치). 마스크 안에 있는 1은 *신경 쓰지 마세요*. 다음 표에서 이 개념을 더 자세히 설명합니다.

마스크의 예	
네트워크 주소(처리할 트래픽)	*10.1.1.0*
mask	*0.0.0.255*
네트워크 주소(바이너리)	*00001010.00000001.00000001.00000000*
마스크(바이너리)	*00000000.00000000.00000000.11111111*

바이너리 마스크를 기준으로 하면, 첫 3개의 세트(옥텟)가 지정된 바이너리 네트워크 주소(00001010.00000001.00000001)와 정확히 일치해야 함을 알 수 있습니다. 마지막 숫자 집합은 *상관 안 함*(.11111111)입니다. 따라서 마지막 옥텟 이후 10.1.1.1로 시작하는 모든 트래픽은 *중요하지 않습니다*. 그러므로 이 마스크를 사용하면 10.1.1.1부터 10.1.1.255(10.1.1.x)까지의 네트워크 주소가 처리됩니다.

ACL 역 마스크를 확인하기 위해 255.255.255.255에서 일반 마스크를 뺍니다. 여기서는 일반 마스크 255.255.255.0을 사용하여 네트워크 주소 172.16.1.0에 대한 역 마스크를 확인합니다.

- 255.255.255.255 - 255.255.255.0 (일반 마스크) = 0.0.0.255 (역 마스크)

ACL 관련 항목을 확인합니다.

- 소스/와일드카드 0.0.0.0/255.255.255.255은 any를 **의미합니다.**
- 10.1.1.2/0.0.0.0의 소스/와일드카드는 **호스트 10.1.1.2와 동일합니다**.

ACL 요약

> **참고:** 서브넷 마스크는 고정 길이 표기법으로도 나타낼 수 있습니다. 예를 들어, 192.168.10.0/24는 192.168.10.0 255.255.255.0입니다.

이 목록에서는 네트워크 범위를 ACL 최적화를 위해 단일 네트워크로 요약하는 방법을 설명합니다. 다음 네트워크를 살펴봅니다.

```
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
```

각 네트워크에서 처음 2개 옥텟 및 마지막 옥텟이 같습니다. 다음 표에서는 이를 단일 네트워크로 요약하는 방법을 설명합니다.

이전 네트워크에 대한 세 번째 옥텟은 각 비트에 대한 옥텟 비트 위치 및 주소 값에 따라 이 표에 표시된 것처럼 쓸 수 있습니다.

10진수 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

앞의 8개 네트워크는 처음 5개 비트가 일치하므로 단일 네트워크로 요약할 수 있습니다 (192.168.32.0/21 또는 192.168.32.0 255.255.248.0). 하위 3개 비트로 가능한 8가지 조합 모두 해당 네트워크 범위에서 유의미합니다. 다음 명령은 이 네트워크를 허용하는 ACL을 정의합니다. 255.255.255.255에서 255.255.248.0(일반 마스크)을 빼면 0.0.7.255가 됩니다.

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

더 자세한 설명을 위해 다음 네트워크 세트를 살펴봅니다.

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

각 네트워크에서 처음 2개 옥텟 및 마지막 옥텟이 같습니다. 다음 표에서는 이를 요약하는 방법을 설명합니다.

이전 네트워크에 대한 세 번째 옥텟은 각 비트에 대한 옥텟 비트 위치 및 주소 값에 따라 이 표에 표시된 것처럼 쓸 수 있습니다.

10진수	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	?	?	?

이전의 예와 달리, 이 네트워크를 단일 네트워크로 요약할 수 없습니다. 단일 네트워크로 요약하면 192.168.144.0/21이 됩니다. 3번째 옥텟에 5개의 비슷한 비트가 있기 때문입니다. 이 요약된 네트워크 192.168.144.0/21은 192.168.144.0부터 192.168.151.0까지의 네트워크 범위를 다룹니다. 이 중 192.168.144.0, 192.168.145.0, 192.168.150.0 및 192.168.151.0 네트워크는 제공된 네 개의 네트워크 목록에 없습니다. 문제의 네트워크를 포괄하려면 2개 이상의 요약 네트워크가 필요합니다. 지정된 4개 네트워크는 다음 2개의 네트워크로 요약할 수 있습니다.

- 네트워크 192.168.146.x 및 192.168.147.x의 경우, 마지막 비트를 제외하고 모든 비트가 일치하는데, 이는 *상관하지 않습니다*. 이를 192.168.146.0/23(또는 192.168.146.0 255.255.254.0)으로 쓸 수 있습니다.
- 네트워크 192.168.148.x 및 192.168.149.x의 경우 마지막 비트를 제외하고 모든 비트가 일치하는데, 이는 *상관없는* 것입니다. 이는 192.168.148.0/23(또는 192.168.148.0 255.255.254.0)으로 쓸 수 있습니다.

이 출력은 이전 네트워크에 대한 요약된 ACL을 정의합니다.

```
!--- 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 처리

라우터에서 수신하는 트래픽을 ACL 항목과 비교하며, 이때 라우터에서 항목이 나타나는 순서를 기준으로 합니다. 새 명령문이 목록의 끝에 추가됩니다. 라우터는 일치하는 항목이 나올 때까지 계속 살펴봅니다. 라우터가 목록의 끝에 도달할 때까지 일치하는 항목이 없을 경우 트래픽이 거부됩니다. 따라서 자주 적중하는 항목이 목록의 맨 위에 있어야 합니다. 허용되지 않은 트래픽에 대한 암시적 거부가 있습니다. 거부 항목이 하나만 있는 단일 항목 ACL은 모든 트래픽을 거부할 수 있습니다. ACL에 하나 이상의 허용 명령문이 있어야 하며, 그렇지 않으면 모든 트래픽이 차단됩니다. 이러한 2가지 ACL(101과 102)은 동일한 효과를 발휘합니다.

```
!--- 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
```

다음 예에서는 마지막 항목으로 충분합니다. IP에는 TCP, UDP(User Datagram Protocol) 및 ICMP(Internet Control Message Protocol)가 포함되므로 처음 세 항목은 필요하지 않습니다.

```
!--- 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
```

포트 및 메시지 유형 정의

ACL 소스 및 대상을 정의할 수 있을 뿐만 아니라 포트, ICMP 메시지 유형 및 기타 매개변수를 정의

할 수도 있습니다. 잘 알려진 포트에 관한 좋은 정보 소스 중 하나가 RFC 1700 입니다. ICMP 메시지 유형은 RFC 792 에서 설명합니다.

라우터는 잘 알려진 일부 포트에 관한 설명 텍스트를 표시할 수 있습니다. 도움말을 보려면?를 사용하십시오.

```
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)
```

컨피그레이션 과정에서 라우터는 숫자 값을 사용자에게 더 편리한 값으로 바꾸기도 합니다. 이것은 ICMP 메시지 유형 번호를 입력하는 예이며 라우터가 번호를 이름으로 변환하도록 합니다.

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

는 다음과 같이 바뀝니다.

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

ACL 적용

ACL을 정의할 수 있지만 적용할 수는 없습니다. 하지만 ACL이 라우터의 인터페이스에 적용되지 않으면 아무런 효과가 없습니다. 트래픽 소스와 가장 가까운 인터페이스에서 ACL을 적용하는 것이 좋습니다. 이 예에서 보여주는 것처럼 소스에서 대상으로의 트래픽을 차단하려고 할 때 라우터 C의 E1에 대한 아웃바운드 목록 대신 라우터 A의 E0에 인바운드 ACL을 적용할 수 있습니다. 액세스 목록은 액세스 목록의 끝에 any any**가 암시적**으로 any **adeny ip**를 가지고 있습니다. 트래픽이 DHCP 요청과 관련된 경우 IP에서 DHCP 요청을 볼 때 소스 주소가 s=0.0.0.0(Ethernet1/0), d=255.255.255.255, len 604, rcvd 2 UDP src=68, dst=67이므로 트래픽이 삭제됩니다. 소스 IP 주소는 0.0.0.0이고 목적지 주소는 255.255.255.255입니다. 소스 포트는 68 및 목적지 67입니다. 따라서 액세스 목록에서 이러한 종류의 트래픽을 허용해야 하며 그렇지 않으면 이 명령문의 끝에 암시적 거부로 인해 트래픽이 삭제됩니다.

참고: UDP 트래픽이 통과하려면 ACL에서 UDP 트래픽도 명시적으로 허용해야 합니다.

입력, 출력, 인바운드, 아웃바운드, 소스, 대상 정의

라우터에서는 입력, 출력, 소스, 대상이라는 용어를 참조로 사용합니다. 라우터의 트래픽은 고속도로의 교통에 비할 수 있습니다. 당신이 펜실베니아에서 법 집행관이었고 메릴랜드에서 뉴욕으로 가는 트럭을 세우고자 했다면, 트럭의 발원지는 메릴랜드이고, 트럭의 목적지는 뉴욕이다. 도로 봉쇄는 펜실베니아-뉴욕 국경(out) 또는 메릴랜드-펜실베니아 국경(in)에서 적용될 수 있다.

라우터에서 각 용어는 다음과 같은 의미를 갖습니다.

- **출력 — 이미 해당 라우터를 지나 인터페이스를 떠나는 트래픽.** 소스는 트래픽이 있었던 위치, 라우터의 반대편이고, 대상은 트래픽의 목적지입니다.
- **입력 — 인터페이스에 도달하여 라우터를 지나는 트래픽.** 소스는 트래픽이 있었던 곳이고, 대상은 트래픽의 목적지, 라우터의 반대편입니다.
- **인바운드 — 액세스 목록이 인바운드일 경우, 라우터가 패킷을 수신하면 Cisco IOS 소프트웨어가 액세스 목록의 조건 명령문을 검토하여 일치하는 항목이 있는지 알아봅니다.** 패킷이 허용될 경우 소프트웨어는 계속해서 패킷을 처리합니다. 패킷이 거부될 경우 소프트웨어는 패킷을 취소합니다.
- **아웃바운드 — 액세스 목록이 아웃바운드일 경우, 소프트웨어에서 패킷을 수신하여 아웃바운드 인터페이스에 라우팅하면 소프트웨어는 액세스 목록의 조건 명령문을 검토하여 일치하는 항목이 있는지 알아봅니다.** 패킷이 허용될 경우 소프트웨어는 패킷을 전송합니다. 패킷이 거부될 경우 소프트웨어는 패킷을 취소합니다.

입력 ACL은 ACL이 적용된 인터페이스의 세그먼트에 소스가, 기타 인터페이스에 대상이 있습니다. 출력 ACL은 ACL이 적용된 인터페이스가 아닌 인터페이스의 세그먼트에 소스가, ACL이 적용된 인터페이스에 대상이 있습니다.

ACL 편집

ACL을 편집하려면 각별한 주의가 필요합니다. 예를 들어, 다음과 같이 번호가 지정된 ACL에서 특정 라인을 삭제하려는 경우, 전체 ACL이 삭제됩니다.

```
!--- 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
```

번호가 지정된 ACL을 편집하기 위해 라우터의 컨피그레이션을 TFTP 서버 또는 텍스트 편집기(예: 메모장)에 복사합니다. 그런 다음 변경하고 다시 컨피그레이션을 라우터로 복사합니다.

다음과 같이 할 수도 있습니다.

```
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
```

삭제 항목은 ACL에서 제거되고, 추가 항목은 ACL의 끝에 생성됩니다.

```
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
```

Cisco IOS에서 일련 번호를 사용하여 ACL 라인을 번호가 지정된 표준 ACL에 또는 번호가 지정된 확장 ACL에 추가할 수도 있습니다. 다음은 이러한 컨피그레이션의 예입니다.

확장 ACL을 다음과 같이 구성합니다.

```
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#
```

ACL **엔트리를** 보려면 theshow access-listcommand를 실행합니다. 10, 20, 30과 같은 일련 번호도 여기에 나타납니다.

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

액세스 목록 101에 일련 번호 5를 사용하여 항목을 추가합니다.

예 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#
```

show access-listcommand 출력에서 시퀀스 번호 5 ACL이 access-list 101의 첫 번째 항목으로 추가됩니다.

```
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#
```

예 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#
```

표준 액세스 목록도 이와 같이 구성할 수 있습니다.

```
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
```

표준 액세스 목록의 주요 차이점은 Cisco IOS에서 시퀀스 번호가 아닌 IP 주소의 하위 순서대로 항목을 추가한다는 것입니다.

다음 예에서는 이를테면 IP 주소(192.168.100.0) 또는 네트워크(10.10.10.0)를 허용하는 방법 등 다양한 항목을 보여줍니다.

```
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
```

IP 주소 172.22.1.1을 허용하기 위해 액세스 목록 2에 항목을 추가합니다.

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

네트워크가 아닌 특정 IP 주소에 우선순위를 주기 위해 이 항목을 목록의 맨 위에 추가합니다.

```
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
```

참고: 이전 ACL은 ASA/PIX 방화벽과 같은 보안 어플라이언스에서 지원하지 않습니다.

액세스 목록을 암호화 맵에 적용할 경우 암호화 목록 변경 지침

- 현재 access-list 컨피그레이션에 추가할 경우 암호화 맵을 제거할 필요가 없습니다. 암호화 맵을 제거하지 않고 곧바로 추가하는 것이 지원되고 허용됩니다.
- 현재 액세스 목록에서 access-list 엔트리를 수정하거나 삭제해야 하는 경우 인터페이스에서 암호화 맵을 제거해야 합니다. 암호화 맵을 제거한 다음 액세스 목록을 변경하고 암호화 맵을 다시 추가합니다. 암호화 맵을 제거하지 않고 액세스 목록 삭제 등의 변경을 하는 것은 지원되지 않으며, 예기치 않은 동작을 유발할 수 있습니다.

문제 해결

인터페이스에서 ACL을 제거하려면 어떻게 해야 합니까?

이 예에서 보여주는 것처럼, 컨피그레이션 모드로 가서 access-group 명령의 앞에 **no를 입력하여 인터페이스에서 ACL을 제거합니다.**

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

너무 많은 트래픽이 거부될 때는 어떻게 해야 합니까?

너무 많은 트래픽이 거부되는 경우, 목록의 로직을 확인하거나 추가로 더 광범위한 목록을 정의하여 적용해보십시오. **show ip access-lists 명령은 어떤 ACL 항목이 적중하는지를 보여주는 패킷 카운트를 제공합니다.** 개별 ACL 항목의 끝에 있는 log 키워드는 포트별 정보와 함께 ACL 번호 및 패킷이 허용 또는 거부되었는지 알려줍니다.

> **참고:** log-input 키워드는 Cisco IOS 소프트웨어 릴리스 11.2 이상 및 통신 사업자 시장을 위해 개발된 특정 Cisco IOS 소프트웨어 릴리스 11.1 기반 소프트웨어에 있습니다. 오래된 소프트웨어는 이 키워드를 지원하지 않습니다. 이 키워드를 사용할 때 입력 인터페이스 및 소스 MAC 주소(해당되는 경우)가 포함됩니다.

Cisco 라우터를 사용하는 패킷 레벨에서 디버깅하려면 어떻게 해야 합니까?

이 절차에서는 디버깅 프로세스를 설명합니다. 시작하기 전에 현재 어떤 ACL도 적용되지 않았고, ACL이 있으며, 빠른 스위칭이 비활성화되지 않았음을 확인합니다.

> **참고:** 트래픽이 많은 시스템을 디버깅할 때는 각별히 주의합니다. 특정 트래픽을 디버깅하려면 ACL을 사용합니다. 그러나 프로세스와 트래픽 흐름을 확인합니다.

1. 원하는 **데이터를** 캡처하려면 access-listcommand를 사용합니다.이 예에서는 대상 주소 10.2.6.6 또는 소스 주소 10.2.6.6에 대해 데이터 캡처가 설정되어 있습니다.
   ```
   access-list 101 permit ip any host 10.2.6.6
   access-list 101 permit ip host 10.2.6.6 any
   ```

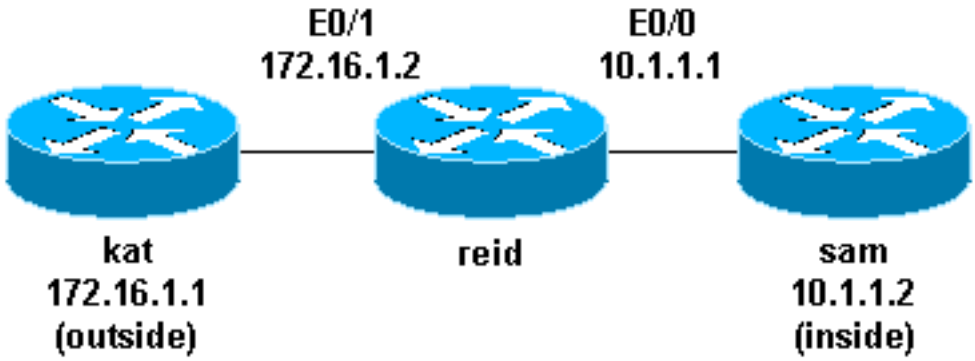
2. 해당 인터페이스에서 빠른 스위칭을 비활성화합니다. 빠른 스위칭이 비활성화되지 않으면 첫 번째 패킷만 표시됩니다.
   ```
   configure terminal
   interface
   ```

3. 활성화 모드에서 **terminal monitor** 명령을 사용하여 현재 터미널 및 세션에 대한 debug 명령 출력 및 시스템 오류 메시지를 표시합니다.
4. 디버그 프로세스를 시작하려면 **debug ip packet 101** 또는 **debug ip packet 101 detail** 명령을 사용합니다.
5. 활성화 모드에서 **no debug all 명령을 실행하고 interface configuration 명령을 실행하여 디버그 프로세스를 중지합니다.**
6. 캐싱을 다시 시작합니다.
   ```
   configure terminal
   interface
   ```

IP ACL의 유형

이 섹션에서는 ACL 유형을 설명합니다.

네트워크 다이어그램



표준 ACL

표준 ACL은 가장 오래된 ACL 유형입니다. Cisco IOS Software Release 8.3으로 거슬러 올라갑니다. 표준 ACL은 IP 패킷의 소스 주소와 ACL에 구성된 주소를 비교하여 트래픽을 제어합니다.

다음은 표준 ACL의 명령문 형식입니다.

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

모든 소프트웨어 릴리스에서 access-*list-number*는 1부터 99까지일 수 있습니다. Cisco IOS 소프트웨어 릴리스 12.0.1에서는 표준 ACL에서 추가 번호(1300~1999)를 사용하기 시작합니다. 이 추가된 번호를 확장 IP ACL이라고 합니다. Cisco IOS 소프트웨어 릴리스 11.2부터 표준 ACL에 목록 *이름을 사용할 수 있게 되었습니다.*

소스/소스 *와일드카드* 설정 0.0.0.0/255.255.255.255은 any로 지정할 수 **있습니다**. 모두 0이라면 와일드카드를 생략할 수 있습니다. 따라서 호스트 10.1.1.2 0.0.0.0은 호스트 10.1.1.2와 동일합니다.

ACL이 정의되면 인터페이스(인바운드 또는 아웃바운드)에 적용해야 합니다. 초기 소프트웨어 릴리스에서는 키워드 out 또는 in이 지정되지 않으면 out이 기본값이었습니다. 나중에 나온 소프트웨어 릴리스에서는 이 방향을 지정해야 합니다.

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

다음은 소스 10.1.1.x에서 온 트래픽을 제외하고 모든 트래픽을 차단하기 위해 표준 ACL을 사용하는 예입니다.

```
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
```

확장된 ACL

확장 ACL은 Cisco IOS Software Release 8.3에 도입되었습니다. 확장 ACL은 IP 패킷의 소스 및 대

상 주소를 ACL에 구성된 주소와 비교하여 트래픽을 제어합니다.

다음은 확장 ACL의 명령문 형식입니다. 공백에 대한 고려를 위해 줄이 여기에 표시됩니다.

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]
```

모든 소프트웨어 릴리스에서 access-*list-number*는 100~199일 수 있습니다. Cisco IOS Software Release 12.0.1에서는 확장 ACL에서 추가 번호(2000~2699)를 사용하기 시작합니다. 이 추가된 번호를 확장 IP ACL이라고 합니다. Cisco IOS 소프트웨어 릴리스 11.2부터 확장 ACL에 목록 *이름을 사용할 수 있게 되었습니다.*

값 0.0.0.0/255.255.255.255는 **any로 지정할 수 있습니다.** ACL이 정의되면 인터페이스(인바운드 또는 아웃바운드)에 적용해야 합니다. 초기 소프트웨어 릴리스에서는 키워드 out 또는 in이 지정되지 않으면 out이 기본값이었습니다. 나중에 나온 소프트웨어 릴리스에서는 이 방향을 지정해야 합니다
.

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

이 확장 ACL은 10.1.1.x 네트워크(내부)의 트래픽을 허용하고 외부로부터 ping 응답을 수신하는 데 사용되며, 외부 사용자로부터의 원치 않는 ping은 차단하여 다른 모든 트래픽을 허용합니다.

```
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
```

> **참고:** 네트워크 관리와 같은 일부 애플리케이션에서는 keepalive 기능에 ping이 필요합니다. 이 경우 차단되거나 허용/거부된 IP에서 더 세분화된 인바운드 ping을 제한할 수 있습니다.

잠금 및 키(동적 ACL)

동적 ACL이라고도 하는 잠금 및 키가 Cisco IOS Software 릴리스 11.1에 도입되었습니다. 이 기능은 텔넷, 인증(로컬 또는 원격) 및 확장 ACL에 따라 달라집니다.

잠금 및 키 컨피그레이션은 라우터를 지나는 트래픽을 차단하기 위해 확장 ACL을 적용하는 것으로 시작합니다. 라우터를 통과하고 싶은 사용자는 라우터에 텔넷하여 인증받지 않으면 확장 ACL에 의해 차단됩니다. 그러면 텔넷 연결이 끊기고 단일 항목 동적 ACL이 존재하는 확장 ACL에 추가됩니다. 그러면 일정 기간 동안 트래픽이 허용됩니다. 유휴 시간 초과 및 절대 시간 초과가 가능합니다.

다음은 로컬 인증의 잠금 및 키 컨피그레이션을 위한 명령문 형식입니다.

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

이 명령의 단일 항목 ACL은 인증 후 기존 ACL에 동적으로 추가됩니다.

```
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
```

다음은 잠금 및 키의 기본적인 예입니다.

```
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
```

10.1.1.2의 사용자가 10.1.1.1에 텔넷 연결하면 동적 ACL이 적용됩니다. 그런 다음 연결이 끊기고 사용자는 172.16.1.x 네트워크로 이동할 수 있습니다.

IP 명명된 ACL

IP 명명된 ACL은 Cisco IOS Software Release 11.2에 도입되었습니다. 이를 통해 표준 및 확장 ACL에 숫자 대신 이름을 지정할 수 있습니다.

다음은 IP 명명된 ACL의 명령문 형식입니다.

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

다음은 TCP의 예입니다.

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

다음은 호스트 10.1.1.2에서 호스트 172.16.1.1로의 텔넷 연결을 제외하고 모든 트래픽을 차단하기 위해 명명된 ACL을 사용하는 예입니다.

```
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
```

재귀 ACL

Reflective ACL은 Cisco IOS Software Release 11.3에 도입되었습니다. Reflective ACL을 사용하면 상위 레이어 세션 정보를 기반으로 IP 패킷을 필터링할 수 있습니다. 일반적으로 라우터 내부에서 발생하는 세션에 대한 응답으로 아웃바운드 트래픽을 허용하고 인바운드 트래픽을 제한하기 위해 사용합니다.

재귀 ACL은 오로지 확장 명명된 IP ACL을 사용하여 정의할 수 있습니다. 번호가 지정된 또는 표준 명명된 IP ACL을 사용하거나 다른 프로토콜 ACL을 사용하여 정의할 수 없습니다. 재귀 ACL은 다른 표준 및 정적 확장 ACL과 함께 사용할 수 있습니다.

다음은 다양한 재귀 ACL 명령문입니다.

```
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>
```

이는 ICMP 아웃바운드 및 인바운드 트래픽 허용의 예이며, 내부에서 시작된 TCP 트래픽만 허용하지만 다른 트래픽은 거부됩니다.

```
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
```

시간 범위를 사용하는 시간 기준 ACL

시간 기준 ACL은 Cisco IOS 소프트웨어 릴리스 12.0.1.T부터 도입했습니다. 확장 ACL in 함수와 비슷하지만, 시간을 기준으로 한 액세스 제어가 가능합니다. 시간 기준 ACL을 구현하기 위해 일일 및 주간의 특정 시간을 정의하는 시간 범위가 생성됩니다. 시간 범위는 이름으로 식별하고, 함수에서 참조합니다. 따라서 시간 제한이 함수 자체에 적용됩니다. 시간 범위는 라우터 시스템 클록을 따릅니다. 라우터 클록을 사용할 수 있으나, NTP(Network Time Protocol) 동기화와 함께 사용하면 가장 효과적입니다.

다음은 시간 기준 ACL 명령입니다.

```
!--- 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
```

이 예에서는 월요일, 수요일, 금요일 업무 시간에 내부 네트워크에서 외부 네트워크로의 텔넷 연결이 허용됩니다.

```
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
```

코멘트 있는 IP ACL 항목

코멘트 있는 IP ACL 항목은 Cisco IOS 소프트웨어 릴리스 12.0.2.T부터 도입했습니다. 코멘트가 있으면 더 쉽게 ACL을 이해할 수 있습니다. 그리고 표준 또는 확장 IP ACL에 사용 가능합니다.

다음은 코멘트가 있는 이름 IP ACL 명령문입니다.

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

다음은 코멘트가 있고 번호가 지정된 IP ACL 명령문입니다.

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

번호가 지정된 ACL 내의 코멘트 예입니다.

```
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
```

상황 기반 액세스 제어

CBAC(Context-Based Access Control)는 Cisco IOS 소프트웨어 릴리스 12.0.5.T부터 도입했습니다. Cisco IOS Firewall 기능 세트가 필요합니다. CBAC는 방화벽을 지나는 트래픽을 검사하여 TCP 및 UDP 세션에 대한 상태 정보를 찾고 관리합니다. 이 상태 정보를 사용하여 방화벽의 액세스 목록에서 임시로 틈새를 만들 수 있습니다. **이**를 위해 허용 가능한 세션, 즉 보호된 내부 네트워크 내에서 시작된 세션에 대해 반환 트래픽 및 추가 데이터 연결을 허용하기 위해 트래픽 시작 흐름의 방향으로 **ip 검사목록**을 구성합니다.

다음은 CBAC 명령문입니다.

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

다음은 아웃바운드 트래픽을 검사하기 위해 CBAC를 사용하는 예입니다. 일반적으로 확장 ACL 111은 CBAC에서 반환 트래픽을 위해 틈새를 만드는 일 없이 ICMP를 제외한 반환 트래픽을 제한합니다.

```
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
```

인증 프록시

인증 프록시는 Cisco IOS 소프트웨어 릴리스 12.0.5.T부터 도입했습니다. Cisco IOS Firewall 기능 세트가 있어야 합니다. 인증 프록시는 인바운드 또는 아웃바운드 사용자 또는 둘 다를 인증하는 데 사용합니다. 일반적으로 ACL에 의해 차단되는 사용자는 브라우저를 시작하여 방화벽을 지나고 TACACS+ 또는 RADIUS 서버에서 인증할 수 있습니다. 서버는 추가 ACL 항목을 라우터까지 전달하여 사용자가 인증 이후 통과할 수 있게 합니다.

인증 프록시는 잠금 및 키(동적 ACL)와 비슷합니다. 다음과 같은 차이점이 있습니다.

- 잠금 및 키는 라우터와의 텔넷 연결에 의해 켜집니다. 인증 프록시는 라우터를 통해 HTTP에 의해 켜집니다.
- 인증 프록시는 외부 서버를 사용해야 합니다.
- 인증 프록시에서는 여러 동적 목록을 추가할 수 있습니다. 잠금 및 키는 하나만 추가할 수 있습니다.
- 인증 프록시는 절대 시간 초과가 있지만 유휴 시간 초과는 없습니다. 잠금 및 키는 둘 다 있습니다.

인증 프록시의 예는 Cisco Secure Integrated Software 컨피그레이션 쿡북을 참조하십시오.

터보 ACL

터보 ACL은 Cisco IOS 소프트웨어 릴리스 12.1.5.T부터 도입했으며, 7200, 7500, 기타 하이엔드 플랫폼에서만 볼 수 있습니다. 터보 ACL 기능은 더 효율적으로 ACL을 처리하여 라우터 성능을 높이기 위해 마련되었습니다.

터보 ACL에는 **access-list compiled 명령을 사용합니다.** 다음은 컴파일된 ACL의 예입니다.

```
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
```

표준 또는 확장 ACL이 정의되면 **global configuration 명령을 사용하여 컴파일합니다.**

```
!--- 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
```

show access-list compiled 명령은 ACL에 관한 통계를 표시합니다.

분산 시간 기준 ACL

분산 시간 기준 ACL은 Cisco IOS 소프트웨어 릴리스 12.2.2.T부터 도입했으며, VPN 기반 7500 시리즈 라우터에서 시간 기준 ACL을 구현하기 위해 마련했습니다. 분산 시간 기준 ACL 기능을 도입하기 전에는 Cisco 7500 시리즈 라우터의 라인 카드에서 시간 기준 ACL을 지원하지 않았습니다. 시간 기준 ACL을 구성했더라도 일반 ACL처럼 작동했습니다. 라인 카드의 인터페이스에서 시간 기준

ACL을 구성한 경우, 인터페이스에 스위칭된 패킷을 라인 카드를 통해 분산 스위칭하지 않고 경로 프로세서로 전달하여 처리했습니다.

분산형 시간 기반 ACL의 구문은 라우트 프로세서와 라인 카드 간의 IPC(Inter Processor Communication) 메시지 상태와 관련하여 명령이 추가되어 시간 기반 ACL의 구문과 동일합니다.

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

수신 ACL

수신 ACL은 라우터의 GRP(gigabit route processor)를 불필요한 잠재적 악성 트래픽으로부터 보호하면서 Cisco 12000 라우터의 보안을 강화하는 데 쓰입니다. 수신 ACL은 Cisco IOS 소프트웨어 릴리스 12.0.21S2에서 유지 관리 조절의 부담을 덜어주는 특별한 기능으로 추가했으며, 12.0(22)S에 통합했습니다. GSR을 참조하십시오. 자세한 내용은 액세스 제어 목록을 참조하십시오.

인프라 보호 ACL

인프라 ACL은 인프라 장비에 대해 승인된 트래픽만 명시적으로 허용하고 다른 모든 전송 트래픽은 허용하여 직접 인프라 공격의 위험과 효과를 최소화하기 위해 사용됩니다. 자세한 내용은 코어 보호: 인프라 보호 액세스 제어 목록을 참조하십시오.

이동 ACL

이동 ACL은 하나 이상의 네트워크에 대한 필요한 트래픽만 명시적으로 허용하여 네트워크 보안을 강화하는 데 사용합니다. 이동 액세스 제어 목록 참조: 에지에서 필터링을 참조하십시오.

관련 정보

- 자주 사용되는 IP ACL 설정
- RFC 1700
- RFC 1918
- 액세스 목록 지원 페이지
- Cisco IOS Firewall
- 기술 지원 및 문서 – Cisco Systems

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.