IPアクセスリストの設定とフィルタリング

内容

概要

このドキュメントでは、さまざまなタイプのIPアクセスコントロールリスト(ACL)と、ネットワークトラフィックをフィルタリングする方法について説明します。

前提条件

要件

このドキュメントに関しては個別の前提条件はありません。このドキュメントで説明しているコンセプトは、Cisco IOS® ソフトウェア リリース 8.3 以降で提供されています。各アクセス リスト機能の注釈を参照してください。

使用するコンポーネント

このドキュメントでは、さまざまなタイプの ACL について説明しています。これらの一部は Cisco IOS ソフトウェア リリース 8.3 から提供されていますが、それより後のソフトウェア リリースで導入されたものもあります。各タイプの説明にある注釈を参照してください。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

表記法

ドキュメント表記の詳細については、『シスコテクニカルティップスの表記法』を参照してください。

背景説明

このドキュメントでは、IP アクセス コントロール リスト（ACL）によるネットワーク トラフィックのフィルタリング方法について説明します。また、IP ACL のタイプについての簡単な説明、機能のアベイラビリティ、およびネットワークでの使用例も示しています。

> **注：**RFC 1700には、既知のポートに割り当てられた番号が含まれています。RFC 1918には、プライベートインターネット用のアドレス割り当て、つまりインターネット上では通常見られないIPアドレスが含まれています。

> **注：**内部情報にアクセスできるのは、登録済みのシスコユーザだけです。

> **注：**ACLは、ネットワークアドレス変換(NAT)へのトラフィックの定義、AppleTalkやIPXなどの非IPプロトコルの暗号化またはフィルタリングにも使用できます。このドキュメントでは、これらの機能については扱っていません。

ACL のコンセプト

マスク

マスクは、IP ACL内のIPアドレスとともに使用され、許可および拒否する必要のある項目を指定します。インターフェイスにIPアドレスを設定するためのマスクは255から始まり、左側に大きな値があります。たとえば、IPアドレス10.165.202.129と255.255.255.224のマスクを使用します。IP ACLのマスクは逆になります（マスク0.0.0.255など）。これは、逆マスクまたはワイルドカードマスクと呼ばれることもあります。マスクの値を2進数（0と1）に分けると、結果によって、トラフィックを処理するときに考慮すべきアドレスビットが決まります。0はの注意が必要なことを示します（完全一致）;マスクの1は*do not care*です。このコンセプトについてさらに詳しく説明するため、次の表の例を使用します。

マスクの例	
ネットワーク アドレス（処理されるトラフィック）	*10.1.1.0*
mask	*0.0.0.255*
ネットワーク アドレス（2 進数）	*00001010.00000001.00000001.00000000*
マスク（2 進数）	*00000000.00000000.00000000.11111111*

2 進数のマスクから、最初の 3 セット（オクテット）が、与えられた 2 進数のネットワーク アドレスと正確に一致する必要があることがわかります（00001010.00000001.00000001）。 最後の数値セットは*無視されます*(.11111111)。 したがって、10.1.1.で始まるすべてのトラフィックは、最後のオクテットが*無視*されるため一致します。その結果、このマスクでは、ネットワーク アドレス 10.1.1.1 ~ 10.1.1.255（10.1.1.x）が処理されます。

ACL 逆マスクを算出するには、255.255.255.255 から通常のマスクを減算します。次の例では、ネットワーク アドレス 172.16.1.0、通常のマスク 255.255.255.0 に対する逆マスクを算出しています。

- 255.255.255.255 - 255.255.255.0（通常のマスク）= 0.0.0.255（逆マスク）

同等のACLに注目してください。

- source/wildcardの0.0.0.0/255.255.255.255はanyを意味し**ます。**
- 10.1.1.2/0.0.0.0の送信元/ワイルドカードは**ホスト10.1.1.2**と同じです。

ACL の集約

> **注：**サブネット マスクは固定長表記でも表現できます。たとえば、192.168.10.0/24 は 192.168.10.0 255.255.255.0 を表します。

続いて、ACL を最適化するために、ある範囲のネットワークを単一のネットワークに集約する方法を説明します。次のネットワークについて考えてみます。

```
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
```

最初の 2 つのオクテットと最後のオクテットはすべてのネットワークで同じです。これらを 1 つのネットワークに集約する方法を次の表に示します。

前のネットワークの3番目のオクテットは、各ビットのオクテットのビット位置とアドレス値に対応して、次の表に示すように記述できます。

10進数	128	64	32	16	8	4	0	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

最初の 5 ビットが一致するので、上記の 8 つのネットワークは 1 つのネットワーク (192.168.32.0/21 または 192.168.32.0 255.255.248.0) に集約できます。 下位 3 ビットの 8 通りの可能な組み合せがすべて、問題のネットワーク範囲に対応します。次のコマンドは、このネットワークを許可する ACL を定義します。255.255.255.255 から 255.255.248.0 (通常のマスク) を差し引くと 0.0.7.255 になります。

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

さらに、次のような一連のネットワークについて考えてみます。

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

最初の 2 つのオクテットと最後のオクテットはすべてのネットワークで同じです。これらを集約する方法を次の表に示します。

前のネットワークの3番目のオクテットは、各ビットのオクテットのビット位置とアドレス値に対応して、次の表に示すように記述できます。

10進数	128	64	32	16	8	4	0	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	?	?	?

前の例とは異なり、これらのネットワークは単一のネットワークに集約できません。単一のネットワークに集約されると、3番目のオクテットに類似した5ビットがあるため192.168.144.0/21になります。この集約ネットワーク192.168.144.0/21は、192.168.144.0 ~ 192.168.151.0の範囲のネットワークをカバーしています。これらのネットワークの中で、192.168.144.0、192.168.150.0、および192.168.151.0のネットワークは、この4つのリストリスト中にありませんネットワークそのネットワークをカバーするには少なくとも二つの集約されたネットワークが必要です。与えられた 4 つのネットワークは、次のように 2 つのネットワークに集約できます。

- ネットワーク192.168.146.xおよび192.168.147.xでは、最後のビットを除くすべてのビットが一致します。最後のビットは*do not care*です。これは、192.168.146.0/23 (または 192.168.146.0 255.255.254.0) と表現できます。

- ネットワーク192.168.148.xおよび192.168.149.xでは、最後のビットを除くすべてのビットが一致します。最後のビットは*do not care*です。これは、192.168.148.0/23（または192.168.148.0 255.255.254.0）と表現できます。

次の出力は、以前のネットワークの集約ACLを定義しています。

```
!--- 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 の処理

ルータに到達したトラフィックは、ACL のエントリと照合されます。照合の順序は、ルータでエントリが生成された順序に従います。新規の文はリストの末尾に追加されます。この照合処理は一致するエントリが見つかるまで続きます。リストの末尾まで照合しても一致するエントリが見つからない場合、そのトラフィックは拒否されます。このため、頻繁にヒットするエントリがリストの先頭に存在する必要があります。許可されないトラフィックについては、黙示的な拒否が適用されます。1つのdenyエントリのみを持つ単一エントリACLは、すべてのトラフィックを拒否できます。ACL に 1 つ以上の permit 文が含まれない場合、すべてのトラフィックがブロックされます。次の 2 つの ACL（101 と 102）は、効果が同じです。

```
!--- 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
```

次の例では、最後のエントリで十分です。IPにはTCP、User Datagram Protocol(UDP)、Internet Control Message Protocol(ICMP)が含まれているため、最初の3つのエントリは必要ありません。

```
!--- 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
```

ポートおよびメッセージ タイプの定義

ACLの送信元と宛先を定義できるだけでなく、ポート、ICMPメッセージタイプ、およびその他のパラメータも定義できます。既知のポートについては、RFC 1700sが情報源として役立ちます。ICMP メッセージ タイプについては、RFC 792 に説明があります。

ルータでは、一部の既知のポートに関する説明を表示できます。ヘルプを表示するには？を使用します。

```
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)
```

また、ルータに数字の値を設定すると、その値がユーザにわかりやすい値に変換されます。次の例では、ICMPメッセージタイプ番号を入力します。これにより、ルータは番号を名前に変換します。

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

これが次のように変換されます。

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

ACL の適用

ACLを定義することはできますが、適用することはできません。しかし、ACL はルータのインターフェイスに適用されない限り効力がありません。ACL は、トラフィックの送信元に最も近いインターフェイスに適用するのがよい方法です。次の例に示すように、送信元から宛先へのトラフィックをブロックしようとする場合は、ルータCのE1への発信リストではなく、ルータAのE0に着信ACLを適用できます。アクセスリストの任意のアクセスリストの最後にadeny ip anyが暗黙的に設定されています。トラフィックがDHCP要求に関連していて、明示的に許可されていない場合、トラフィックは廃棄されます。これは、IPでDHCP要求を見ると、送信元アドレスがs=0.0.0.0 (Ethernet1/0)、d=255.255.255、len 604、rcvd 2 UDP src=68、dst=67であるためです。送信元IPアドレスが0.0.0.0で、宛先アドレスが2555.25555.255 55.送信元ポートは68、宛先ポートは67です。したがって、アクセスリストでこの種のトラフィックを許可する必要があります。許可しないと、文の最後の暗黙の拒否によってトラフィックがドロップされます。

> **注**：UDPトラフィックが通過するためには、UDPトラフィックもACLによって明示的に許可される必要があります。

in、out、着信、発信、送信元、および宛先の定義

ルータでは、基準として in、out、source（送信元）、および destination（宛先）という用語が使用されます。ルータ上のトラフィックは、高速道路のトラフィックにたとえることができます。ペンシルベニア州の法執行官が、メリーランド州からニューヨーク州に移動するトラックを停止する場合、トラックの送信元はメリーランド州で、トラックの宛先はニューヨーク州です。この道路ブロックは、ペンシルベニア州とニューヨーク州の境界(out)またはメリーランド州とペンシルベニア州の境界(in)に適用できます。

ルータの場合、これらの用語の意味は次のようになります。

- **Out**：すでにルータを通過し、インターフェイスから送出されるトラフィック。トラフィックの元の場所（ルータのもう一方の側）が送信元で、トラフィックが向かっている場所が宛先です。
- **In**：現在インターフェイスに到達していて、これからルータを通過するトラフィック。トラフィックの元の場所が送信元で、トラフィックが向かっている場所（ルータのもう一方の側）が宛先です。
- **インバウンド：アクセス リストが着信する場合、ルータがパケットを受信すると、Cisco IOSソフトウェアは、一致がアクセス リスト ステートメント条件を確認します。**パケットが許可されている場合、ソフトウェアはパケットの処理を続行します。パケットが拒否されると、パケットは廃棄されます。
- **アウトバウンド：アクセス リストがアウトバウンドの場合、ソフトウェアがインターフェイスにパケットを受信し、ルーティングすると、ソフトウェアは合致するようアクセス リスト ステートメント条件を確認します。**パケットが許可された場合、ソフトウェアはパケットを送信します。パケットが拒否されると、パケットは廃棄されます。

in ACL の場合、送信元は適用インターフェイスのセグメント上にあり、宛先はそれ以外のインターフェイスの先にあります。out ACL の場合、送信元は適用インターフェイス以外のインターフェイスのセグメント上にあり、宛先は適用インターフェイスの先にあります。

ACL の編集

ACL を編集するときには、特別な注意が必要です。たとえば、次のように既存の番号付き ACL から特定の行を削除しようとすると、その ACL 全体が削除されてしまいます。

```
!--- 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
```

番号付き ACL を編集するには、ルータの設定を TFTP サーバ、またはメモ帳などのテキスト エディタにコピーします。次に、変更を加えてから、設定をルータにコピーします。

次のようにして編集することも可能です。

```
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
```

削除したエントリは ACL から除去され、追加したエントリは ACL の末尾に追加されます。

```
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
```

番号付きの標準 ACL や拡張 ACL に、Cisco IOS 内のシーケンス番号により ACL 行を追加することもできます。次に設定の例を示します。

次のように拡張 ACL を設定します。

```
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エントリを表示するには、show access-listコマンドを発行します。出力には、10、20、30などのシーケンス番号も表示されます。

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

シーケンス番号 5 のエントリをアクセス リスト 101 に追加します。

例 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-listコマンド出力では、シーケンス番号5 ACLがアクセスリスト101の最初のエントリとして追加されています。

```
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#
```

例 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#
```

同様に、次のように標準アクセス リストを設定できます。

```
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
```

標準アクセスリストの主な違いは、Cisco IOSがシーケンス番号ではなくIPアドレスの子孫順にエントリを追加する点です。

たとえば、次の例は、IP アドレス 192.168.100.0 またはネットワーク 10.10.10.0 を許可する方法を示しています。

```
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 アドレス 172.22.1.1 を許可するために、アクセス リスト 2 にエントリを追加します。

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

ネットワークよりも特定の IP アドレスを優先するために、エントリをリストの上部に追加します。

```
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
```

注：ASA/PIX ファイアウォールなどのセキュリティ アプライアンスでは、上記 ACL はサポートされていません。

クリプトマップに適用されるアクセス リスト変更のガイドライン

- 現在のアクセスリスト設定に追加する場合は、暗号マップを削除する必要はありません。クリプトマップを削除しないで設定を直接追加することはサポートされており、許容されます。
- 現在のアクセスリストからアクセスリストエントリを変更または削除する必要がある場合は、インターフェイスからクリプトマップを削除する必要があります。クリプトマップを削除した後、アクセス リストですべての変更を行ってから、クリプトマップを再度追加します。クリプトマップを削除しないでアクセス リストを削除するなどの変更を行うことはサポートされておらず、予測できない動作を引き起こす可能性があります。

トラブルシュート

ACL をインターフェイスから削除するにはどうすればいいですか。

ACL をインターフェイスから削除するには、設定モードで **access-group コマンドの前に no を入力します。次の例を参照してください。**

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

拒否されるトラフィックが多過ぎる場合はどうすればいいですか。

拒否されるトラフィックが多過ぎる場合は、リストのロジックについて検討するか、または新たにより範囲の広いリストを定義して適用してみます。**show ip access-lists コマンドを使用すれば、ヒットしている ACL エントリを示すパケット カウントを表示できます。**各 ACL エントリの末尾に log キーワードを使用すると、ポート固有の情報以外に、ACL 番号と、パケットが許可されたか拒否されたかが表示されます。

注：log-input キーワードが存在するのは、Cisco IOS ソフトウェア リリース 11.2 以降と、サービス プロバイダー市場向けに特別に作成された Cisco IOS ソフトウェア リリース 11.1 ベースのソフトウェアです。古いソフトウェアでは、このキーワードがサポートされません。このキーワードの用途には、入力インターフェイスと送信元 MAC アドレス (該当する場合) も含まれます。

Cisco ルータを使用して、パケット レベルのデバッグを行うにはどうすればいいですか。

デバッグを実行する手順は次のとおりです。始める前に、いずれの ACL も現在適用されていないこと、ACL が存在すること、およびファースト スイッチングが無効になっていないことを確認します。

注：大量のトラフィックが流れるシステムをデバッグする際は、細心の注意が必要です。1 つの ACL を使用して特定のトラフィックをデバッグします。プロセスとトラフィックフローを確認します。

1. 目的のデータをキャプチャするには、access-listコマンドを使用します。次の例では、データ キャプチャが宛先アドレス 10.2.6.6 または送信元アドレス 10.2.6.6 に設定されています。
```
access-list 101 permit ip any host 10.2.6.6
access-list 101 permit ip host 10.2.6.6 any
```

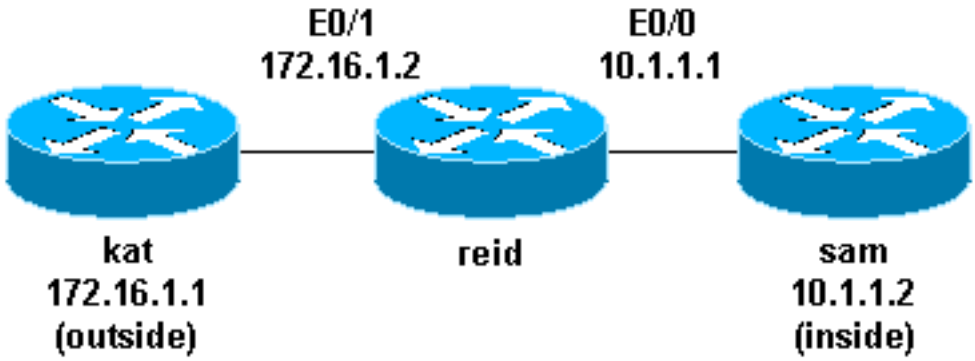
2. 関係するインターフェイスのファースト スイッチングを無効にします。ファースト スイッチングが無効になっていない場合、最初のパケットのみが表示されます。
```
configure terminal
interface
```

3. イネーブル モードで terminal monitor コマンドを使用して、現在のターミナルおよびセッションに関する debug コマンド出力およびシステム エラー メッセージを表示します。
4. デバッグプロセスを開始するには、debug ip packet 101またはdebug ip packet 101 detailコマンドを使用します。
5. イネーブル モードで no debug all コマンドを実行し、interface configuration コマンドを実行して、デバッグ プロセスを停止します。
6. キャッシングを再開します。
```
configure terminal
interface
```

IP ACL のタイプ

この項では、ACL のタイプについて説明します。

ネットワーク図



標準 ACL

標準 ACL は最も古いタイプの ACL で、Cisco IOSソフトウェアリリース8.3の初期のバージョンに遡ります。標準ACLは、IPパケットの送信元アドレスとACLで設定されたアドレスを比較することによって、トラフィックを制御します。

標準 ACL のコマンド構文形式を次に示します。

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

すべてのソフトウェアリリースで、*access-list-number*には1から99までの任意の値を指定できます。Cisco IOSソフトウェアリリース12.0.1では、標準ACLが追加の番号(1300 ~ 1999)を使用し始めます。 これらの追加の番号は、拡張 IP ACL と呼ばれます。Cisco IOS ソフトウェア リリース 11.2 では、標準 ACL でリストの*名前を使用する機能が追加されました。*

*source/source-wildcard*の設定は、**any**として0.0.0.0/255.255.255.255を指定できます。ワイルドカードは、すべてゼロの場合は省略できます。したがって、ホスト10.1.1.2 0.0.0.0はホスト10.1.1.2と同じです。

ACL を定義した後、インターフェイス (着信または発信) に適用する必要があります。 初期のソフトウェア リリースでは、キーワード out または in が指定されていない場合は、out がデフォルトでした。新しいソフトウェア リリースでは、方向を必ず指定する必要があります。

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

次に、送信元が 10.1.1.x 以外のトラフィックをすべてブロックする標準 ACL の使用例を示します。

```
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
```

拡張 ACL

拡張ACLは、Cisco IOSソフトウェアリリース8.3で導入されました。拡張ACLは、IPパケットの送信元アドレスと宛先アドレスをACLで設定されたアドレスと比較することによって、トラフィックを制御します。

拡張 ACL のコマンド構文形式を次に示します。スペースを考慮して、ここでは行を折り返します。

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]
```

すべてのソフトウェアリリースで、*access-list-number*には100 ~ 199を指定できます。Cisco IOSソフトウェアリリース12.0.1では、拡張ACLが追加の番号(2000 ~ 2699)を使用し始めます。これらの追加の番号は、拡張 IP ACL と呼ばれます。Cisco IOS ソフトウェア リリース 11.2 では、拡張 ACL でリストの*名前を使用する機能が追加されました。*

値 0.0.0.0/255.255.255.255 は **any として指定できます。**ACL を定義した後、インターフェイス（着信または発信）に適用する必要があります。 初期のソフトウェア リリースでは、キーワード out または in が指定されていない場合は、out がデフォルトでした。新しいソフトウェア リリースでは、方向を必ず指定する必要があります。

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

この拡張ACLは、10.1.1.xネットワーク(inside)上のトラフィックを許可し、outsideからのping応答を受信する一方で、outsideのユーザからの要求外のpingを拒否して、他のすべてのトラフィックを許可するために使用されます。

```
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
```

> **注：**ネットワーク管理などの一部のアプリケーションでは、キープアライブ機能のために PING が必要です。この場合、ブロックされる着信pingを制限するか、許可/拒否されるIPをより細かく設定できます。

ロック アンド キー（ダイナミック ACL）

ロックアンドキー（ダイナミックACLとも呼ばれる）は、Cisco IOSソフトウェアリリース11.1で導入されました。この機能は、Telnet、認証（ローカルまたはリモート）、および拡張ACLに依存します。

ロック アンド キーの設定は、ルータを通過するトラフィックに拡張 ACL を適用してブロックすることから始まります。ルータに接続しようとするユーザは、Telnet を使ってルータに接続し、認証されるまでエクステンデッド（拡張）ACL によってブロックされる。その後、Telnet接続が切断され、存在する拡張ACLに単一エントリのダイナミックACLが追加されます。この割り当ては、特定の期間中、または;アイドルおよび絶対タイムアウトがあります。

次に、ローカル認証でロック アンド キーを設定するためのコマンド構文形式を示します。

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

次のコマンドに含まれる単一エントリの ACL は、認証後、既存の ACL にダイナミックに追加されます。

```
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
```

次に、ロック アンド キーの基本的な例を示します。

```
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
```

10.1.1.2 のユーザが 10.1.1.1 への Telnet 接続を実行すると、ダイナミック ACL が適用されます。続いてこの接続が解除され、このユーザが 172.16.1.x ネットワークにアクセスできるようになります。

IP 名前付き ACL

IP名前付きACLは、Cisco IOSソフトウェアリリース11.2で導入されました。これにより、標準ACLと拡張ACLに、番号ではなく名前を付けることができます。

IP 名前付き ACL のコマンド構文形式を次に示します。

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

次に TCP の例を示します。

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

次に、名前付き ACL を使用して、ホスト 10.1.1.2 からホスト 172.16.1.1 への Telnet 接続以外のトラフィックをすべてブロックする例を示します。

```
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
```

再帰 ACL

再帰ACLは、Cisco IOSソフトウェアリリース11.3で導入されました。再帰ACLを使用すると、上位層のセッション情報に基づいてIPパケットをフィルタリングできます。一般に再帰 ACL は、ルータ内部から開始されたセッションに対して、発信トラフィックを許可し着信トラフィックを制限するために使用されます。

再帰 ACL は、拡張名前付き IP ACL でのみ定義できます。番号付きまたは標準名前付き IP ACL、またはその他のプロトコル ACL では定義できません。再帰 ACL は、他の標準 ACL やスタティックな拡張 ACL と組み合せて使用できます。

次に、さまざまな再帰 ACL コマンドの構文を示します。

```
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>
```

これは、ICMPの発信および着信トラフィックの許可の例ですが、内部から開始されたTCPトラフィックだけが許可され、他のトラフィックは拒否されます。

```
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
```

時間範囲を使用する時間ベース ACL

時間ベース ACL は、Cisco IOS ソフトウェア リリース 12.0.1.T で導入されました。機能的には拡張 ACL に似ていますが、時間に基づくアクセス制御が可能です。時間ベース ACL を実装するには、日および曜日の特定の時間を指定する時間範囲を作成します。この時間範囲は名前によって識別され、次に機能によって参照されます。したがって、時間制限は機能自体に課されます。時間範囲はルータのシステム クロックに基づきます。ルータのクロックも使用できますが、この機能は Network Time Protocol (NTP; ネットワーク タイム プロトコル) 同期を併用した場合に最適に動作します。

次に時間ベース ACL のコマンドを示します。

```
!--- 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
```

次の例では、内部ネットワークから外部ネットワークへの Telnet 接続が月、水、および金曜日の業務時間内に許可されます。

```
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
```

コメント付き IP ACL エントリ

コメント付き IP ACL エントリは、Cisco IOS ソフトウェア リリース 12.0.2.T で導入されました。コメントにより、ACL が理解しやすくなります。コメントは標準または拡張 IP ACL に使用できます。

次に、コメント付きの名前付き IP ACL コマンドの構文を示します。

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

次に、コメント付きの番号付き IP ACL コマンドの構文を示します。

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

これは、番号付きACL内のコメントの例です。

```
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
```

コンテキストベース アクセス制御

Context-based Access Control (CBAC; コンテキストベース アクセス制御) は、Cisco IOS ソフトウェア リリース 12.0.5.T で導入されました。CBAC には Cisco IOS Firewall フィーチャ セットが必要です。CBAC は、ファイアウォールを通過するトラフィックを調べ、TCP および UDP セッションのステート情報の検出と管理を行います。このステート情報は、ファイアウォールのアクセス リストに一時的な開口部を作成するために使用されます。**トラフィック開始フローの方向でip inspectlists**を設定して、リターントラフィックと、許可されるセッション (保護された内部ネットワーク内から開始されるセッション) に対する追加のデータ接続を許可します。

次に CBAC の構文を示します。

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

次に、CBAC を使用して発信トラフィックを調べる例を示します。拡張 ACL 111 は、CBAC によってリターン トラフィック用の開口部が空けられていなければ、通常は ICMP 以外のリターン トラフィックをブロックします。

```
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
```

認証プロキシ

認証プロキシは、Cisco IOS ソフトウェア リリース 12.0.5.T で導入されました。認証プロキシには、Cisco IOS Firewall フィーチャ セットが必要です。認証プロキシは、着信または発信ユーザ、もしくはその両方を認証するために使用します。通常は ACL によってブロックされるユーザが、ブラウザを起動してファイアウォールを通過し、TACACS+ または RADIUS サーバで認証を受けることができます。認証されたユーザが通過できるように、サーバからルータに追加の ACL エントリが渡されます。

認証プロキシはロック アンド キー (ダイナミック ACL) と似ています。 ただし、次の点が異なります。

- ロック アンド キーはルータへの Telnet 接続によってオンになります。認証プロキシはルータを経由する HTTP によってオンになります。
- 認証プロキシには外部サーバを使用する必要があります。
- 認証プロキシでは複数のダイナミック リストの追加を処理できます。ロック アンド キーで追加できるのは 1 つだけです。
- 認証プロキシには絶対タイムアウトはありますが、アイドル タイムアウトはありません。ロック アンド キーには両方のタイムアウトがあります。

認証プロキシの例については、『Cisco セキュリティ統合ソフトウェア設定クックブック』を参照してください。

ターボ ACL

ターボ ACL は、Cisco IOS ソフトウェア リリース 12.1.5.T で導入されました。7200、7500、およびその他のハイエンド プラットフォームでのみ使用されています。ターボ ACL 機能は、ACL 処理の効率化によってルータのパフォーマンスを向上させる目的で設計されています。

ターボ ACL には **access-list compiled コマンドを使用します。**コンパイルされた ACL の例を次に示します。

```
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
```

標準または拡張 ACL を定義した後、**global configuration コマンドを使用してコンパイルします。**

```
!--- 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 コマンドは、ACL に関する統計情報を表示します。

分散型時間ベース ACL

分散型時間ベース ACL は、VPN 対応 7500 シリーズ ルータに時間ベース ACL を実装するため、Cisco IOS ソフトウェア リリース 12.2.2.T で導入されています。分散型時間ベース ACL 機能の登場以前は、Cisco 7500 シリーズ ルータ用ライン カードでは時間ベース ACL が サポートされていませんでした。時間ベース ACL が設定されている場合は、通常の ACL として動作していました。ラインカード上のインターフェイスに時間ベース ACL が設定されている場合、そのインターフェイスにスイッチされたパケットは、そのラインカードを通じて分散スイッチングされず、処理のためにルート プロセッサに転送されていました。

分散型時間ベースACLの構文は、ルートプロセッサとラインカード間のInter Processor Communication（IPC；プロセッサ間通信）メッセージのステータスに関するコマンドが追加されている点を除き、時間ベースACLの構文と同じです。

```
debug time-range ipc
```

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

受信 ACL

受信 ACL は、弊害を含む可能性のある不必要なトラフィックからルータの Gigabit Route Processor (GRP; ギガビット ルート プロセッサ) を保護することにより、Cisco 12000 ルータのセキュリティを強化するために使用されます。受信 ACL は、Cisco IOS ソフトウェア リリース 12.0.21S2 では特別なメンテナンスとして追加されていましたが、12.0(22)S に統合されました。詳細については、『GSR:受信アクセスコントロールリスト』を参照してください。

インフラストラクチャ保護 ACL

インフラストラクチャACLは、インフラストラクチャ機器に対する許可されたトラフィックだけを明示的に許可し、その他すべての中継トラフィックは許可することによって、インフラストラクチャへの直接的な攻撃のリスクと効果を最小限に抑えるために使用されます。コアを保護することを参照:詳細については、インフラストラクチャ保護のアクセス コントロール リスト。

トランジット ACL

トランジット ACL は、ネットワークへの必要なトラフィックだけを明示的に許可するので、ネットワークのセキュリティを強化するために使用されます。中継アクセス コントロール リストを参照:最後の詳細については、フィルタリング。

関連情報

- 一般的に使用される IP ACL の設定
- RFC 1700
- RFC 1918
- アクセス リストに関するサポートページ
- Cisco IOS ファイアウォール
- テクニカル サポートとドキュメント – Cisco Systems

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。