CHAPTER 19

スタンドアロン Content Engine での IP アクセス制御リストの作成と管理

この章では、スタンドアロン Content Engine で IP Access Control List（ACL; アクセス制御リスト）を作成し、管理する方法について説明します。この章の内容は、次のとおりです。

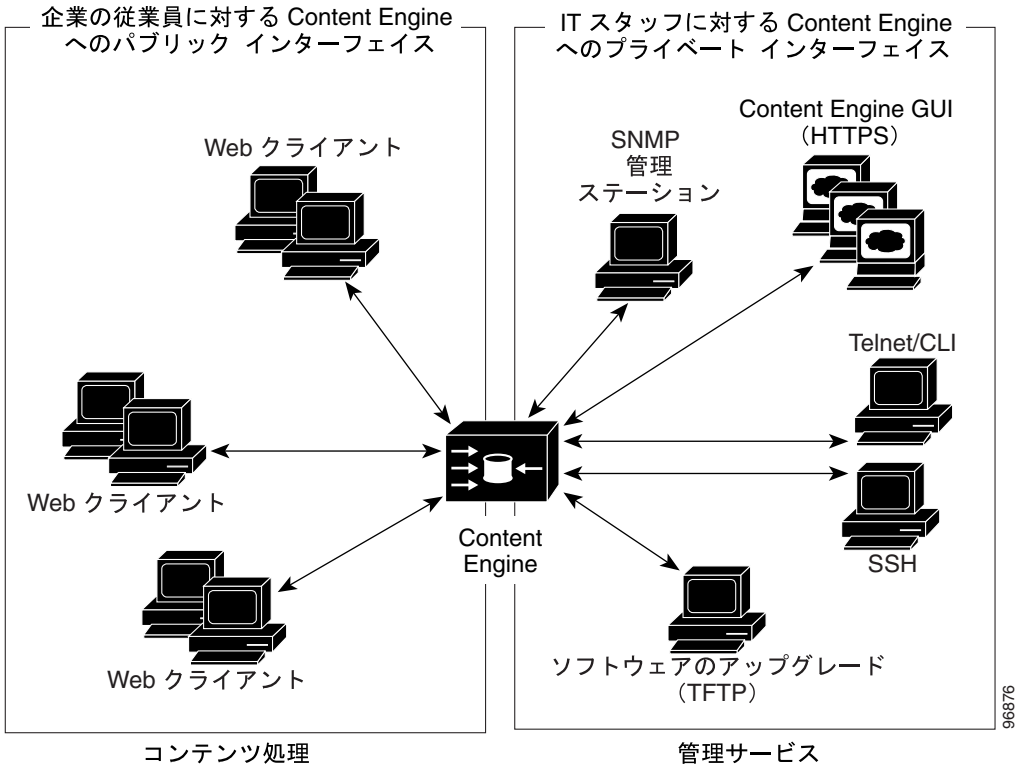
（注） この章で使用されている *IP ACL* という用語は、IP アクセス制御リストを表しています。

スタンドアロン Content Engine 用の IP ACL の導入

ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、IP ACL がサポートされています。IP ACL は IP パケット フィルタリングを提供します。IP ACL では、Content Engine の特定のインターフェイスを IP パケットが通過するのを許可または拒否することによって、パケットをフィルタリングできます。

スタンドアロン Content Engine を配置した環境では、この機能を使用して Content Engine 上でコンテンツ サービスと管理サービスへのアクセスをコントロールできます。たとえば、IP ACL を使用して、コンテンツ配信に対する Content Engine のパブリック インターフェイスと、管理サービス（たとえば、Telnet、Secure Shell［SSH；セキュア シェル］、SNMP、HTTPS、ソフトウェア アップグレードなど）用のプライベート インターフェイスを定義することができます（図 19-1 を参照）。

図 19-1　IP ACL によるスタンドアロン Content Engine の特定インターフェイスへのアクセスのコントロール



（注）　ACNS 5.4.1 ソフトウェアおよびそれ以降のリリースでは、IP ACL は非透過（プロキシ モード要求）着信 FTP ネイティブ要求および透過的にリダイレクトされる着信 FTP ネイティブ要求でもサポートされます。詳細は、「IP ACL による ネイティブ FTP アクセスのコントロール」（p.19-20）を参照してください。

次に示すのは、スタンドアロン Content Engine を配置した環境での IP ACL の使用例です。

- Content Engine が顧客の建物内にあり、サービス プロバイダーによって管理され、サービス プロバイダーは管理のためにデバイスを保護したいと考えている。

- Content Engine　が企業内のあらゆる場所に配置されている。ルータやスイッチと同様、管理者は Telnet、SSH、Content Engine GUI による IT ソース サブネットへのアクセスを制限したいと考えている。
- 強固な外部インターフェイスを備えたアプリケーション レイヤ プロキシ ファイアウォールに危険なポートがない（*強固*とは、セキュリティ上、インターフェイスによって主にどのポートをアクセス用として使用可能にするかを厳しく制限することを意味します。外部インターフェイスがあると、さまざまなタイプのセキュリティ攻撃が可能になります）。Content Engine の外部アドレスはインターネット グローバルであり、内部アドレスはプライベートです。内部インターフェイスには、Content Engine に対する Telnet、SSH、Content Engine GUI アクセスを制限するための IP ACL があります。
- 信頼できない環境で、Content Engine がリバース プロキシとして配置されている。Content Engine 管理者は、バックエンド インターフェイス上の外部インターフェイスと発信接続でポート 80 の着信トラフィックのみの許可を考えます。
- WCCP を使用した Content Engine がファイアウォールとインターネット ルータ、またはインターネット ルータから離れたサブネットの間に置かれている。Content Engine とルータの両方に IP ACL が必要です。

スタンドアロン Content Engine 用の IP ACL の導入

IP ACL を実装する手順は、次のとおりです。

ステップ 1　**ip access-list** コマンドを使用して、IP ACL をスタンドアロン Content Engine で定義します。

ステップ 2　**ip access-group** コマンドを使用して、着信または発信の定義済み IP ACL をスタンドアロン Content Engine のインターフェイスに適用します。

（注） IP ACL は、このスタンドアロン Content Engine への Telnet、SSH、SNMP によるアクセスに対する許可または拒否にも使用できます。

IP ACL の定義およびアクティブ化の例

次の例では、スタンドアロン Content Engine で IP ACL を定義し、アクティブにする方法を示します。この例に示すように、最初に **ip access-list** グローバル コンフィギュレーション コマンドを使用して、スタンドアロン Content Engine 用の IP ACL を作成します。この場合、IP ACL を example と名付けて、すべての Web トラフィックを許可している一方、ある特定のホストへの SSH アクセスは制限しています。

```
ContentEngine(config)# ip access-list extended example
ContentEngine(config-ext-nacl)# permit tcp any any eq www
ContentEngine(config-ext-nacl)# permit tcp host 64.101.215.21 any eq ssh
ContentEngine(config-ext-nacl)# exit
```

IP ACL を作成したら、**interface** グローバル コンフィギュレーション コマンドと **ip access-group** コンフィギュレーション インターフェイス コマンドを使用して、Content Engine の特定のインターフェイスに対して IP ACL を適用し、アクティブにします。

```
ContentEngine(config)# interface gigabitethernet 1/0
ContentEngine(config-if)# ip access-group example in
ContentEngine(config-if)# exit
```

IP ACL を定義してアクティブにしたら、Content Engine での実行コンフィギュレーションを表示します。

```
ContentEngine# show running-config
.
.
.
!
interface GigabitEthernet 1/0
 ip address 10.1.1.50 255.255.0.0
 ip access-group example in
 exit
.
.
.
ip access-list extended example
 permit tcp any any eq www
 permit tcp host 10.101.215.21 any eq ssh
 exit

.

.

.
```

（注）　IP ACL は、各 ACNS ソフトウェア デバイスに対してのみ定義されます。IP ACL は、ACNS ネットワーク全体に対して、またはデバイス グループを介してグローバルに管理することはできません。Content Distribution Manager による ACNS ネットワーク デバイス（たとえば、Content Distribution Manager に登録している Content Engine など）での IP ACL の作成および管理に関する詳細は、『*Cisco ACNS Software Configuration Guide for Centrally Managed Deployments*』Release 5.5 を参照してください。

IP ACL の基本的な情報については、次の「IP ACL 使用の概要」を参照してください。IP ACL の設定方法については、「スタンドアロン Content Engine での IP ACL の定義とアクティブ化」（p.19-7）を参照してください。

IP ACL 使用の概要

IP ACL は、1 つまたは複数の条件エントリから構成されます。これらのエントリは、Content Engine が以降の処理のために廃棄または承認するパケットのタイプを指定します。Content Engine はそれぞれの条件を IP ACL にある順序で適用します。このデフォルトは、ユーザが条件を設定した順序です。

ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、次の 2 つのタイプの IP ACL があります。

- 標準（Standard）ACL
- 拡張（Extended）ACL

（注）　スタンドアロン Content Engine で IP ACL を作成し、管理する場合には、ACNS ソフトウェア CLI を使用する必要があります。Content Engine GUI は、現時点では、スタンドアロン Content Engine 上での IP ACL の設定をサポートしていません。

標準 IP ACL の使用

標準 ACL は通常、次の目的に使用されます。

- 特定の IP アドレスを持つホストからの接続を許可する。
- 特定のネットワーク上のホストからの接続を許可する。

標準 IP ACL コンフィギュレーション モードへのアクセス

標準 IP ACL を使用するには、Content Engine 上で標準 IP ACL コンフィギュレーション モードに入る必要があります。標準 IP ACL コンフィギュレーション モードにアクセスするには、**ip access-list standard** グローバル コンフィギュレーション コマンドを入力します。

```
ContentEngine(config)# ip access-list standard {acl-name | acl-num}
```

- *acl-name* は、作成または変更する標準 IP ACL の名前です。
- *acl-num* は、作成または変更する標準 IP ACL の番号です。

標準 IP ACL モードに入ると、ContentEngine(config)# プロンプトが ContentEngine(config-std-nacl)# に変更されます。この場合の nacl は、指定の標準アクセス リストです。

たとえば、次の例では、ACL 番号 2 の標準 IP ACL を変更するために標準 IP ACL コンフィギュレーション モードに入る方法を示しています。この CLI は標準 IP ACL コンフィギュレーション モードに入り、このモードでは、以降のすべてのコマンドが現在指定されている標準 IP ACL（たとえば、標準 IP ACL *nac2*）に適用されます。

```
ContentEngine(config)# ip access-list standard 2
ContentEngine(config-std-nacl)#
```

拡張 IP ACL の使用

拡張 IP ACL では通常、接続のコントロールで次のエレメントを使用します。

- 送信先 IP アドレス
- IP プロトコル タイプ
- UDP または TCP の送信元ポート、あるいは送信先ポート
- ICMP メッセージのタイプまたはコード
- TCP フラッグ ビット（設定済み）

さらに制限された条件を作成するには、これらの条件を送信元 IP アドレスの情報と組み合わせることができます。表 19-3 は、特定の Internet Control Message Protocol（ICMP）メッセージのタイプとコードの照合に使用できるキーワードを示しています。

拡張 IP ACL コンフィギュレーション モードへのアクセス

拡張 IP ACL を使用するには、Content Engine 上で拡張 IP ACL コンフィギュレーション モードに入る必要があります。拡張 IP ACL コンフィギュレーション モードにアクセスするには、**ip access-list extended** グローバル コンフィギュレーション コマンドを入力します。

```
ContentEngine(config)# ip access-list extended {acl-name | acl-num}
```

- *acl-name* は、作成または変更する拡張 IP ACL の名前です。
- *acl-num* は、作成または変更する拡張 IP ACL の番号です。

拡張 IP ACL モードに入ると、ContentEngine(config)# プロンプトが ContentEngine(config-ext-nacl)# プロンプトに変更されます。ここで nacl は、指定の拡張アクセス リストを表します。

次の例では、ACL 番号 101 の拡張 IP ACL を変更するために、拡張 IP ACL コンフィギュレーション モードに入る方法を示しています。この CLI は拡張 IP ACL コンフィギュレーション モードに入り、このモードでは、以降のすべてのコマンドが現在指定されている拡張 IP ACL（たとえば、拡張 IP ACL 101）に適用されます。

```
ContentEngine(config)# ip access-list extended 101
ContentEngine(config-ext-nacl)#
```

（注）　拡張 IP ACL を作成または変更する方法については、「スタンドアロン Content Engine での IP ACL の作成または変更」（p.19-11）を参照してください。

スタンドアロン Content Engine での IP ACL の定義とアクティブ化

一部のサービス プロバイダーの配置では、Content Engine に、コンテンツを配信するためのユーザの IP アドレス スペース内のインターフェイスと、管理者が管理用に使用するプライベート IP アドレス スペース内のインターフェイスを指定できます。ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、さまざまなサービスを特定のインターフェイスに（たとえば、管理サービスをプライベート IP アドレス スペースに）関連付けることができます。これにより、企業のユーザは管理目的ではなく、コンテンツ配信だけのために Content Engine にアクセスできます。

ACNS 5.1 ソフトウェアおよびそれ以降のリリースを実行しているスタンドアロン Content Engine を配置した環境下で IP ACL を使用するには、システム管理者は CLI を使用して次の作業を行う必要があります。

1. **ip access-list** コマンドを使用して、IP ACL を定義します。
2. **interface** と **ip access-group** の各コマンドを使用して、IP ACL を Content Engine の特定のインターフェイスに適用します。

ヒント　IP ACL をインターフェイス上の着信または発信 IP トラフィックに適用するには、**ip access-group** コマンドを使用します。

使用上の注意事項

スタンドアロン Content Engine で IP ACL を作成または変更する際は、次の重要な点に留意してください。

- 標準または拡張 IP ACL にエントリを作成するには、**deny** または **permit** キーワードを使用し、Content Engine が以降の処理のために廃棄または承認するパケットのタイプを指定します。デフォルトでは、アクセス リストによりすべてが拒否されます。これは、このリストが暗黙の **deny any** エントリによって終了されるためです。したがって、有効なアクセス リストを作成するには、少なくとも 1 つの **permit** エントリを指定する必要があります。

  （注）　特定のネットワークからの接続を許可するには、**permit** *source-ip wildcard* コマンドを使用します。*source-ip* には、指定するネットワーク上のホストのネットワーク ID または IP アドレスを入力します。*wildcard* には、サブネット マスクのリバースであるマスクのドット付き 10 進表記を入力します。このワイルドカードでは、0 は一致させるべき位置を表し、1 は何も特定しない位置を表します。たとえば、ワイルドカード 0.0.0.255 では送信元 IP アドレスの末尾の 8 ビットが無視されます。したがって、**permit 192.168.1.0 0.0.0.255** のエントリは、192.168.1.0 ネットワーク上のすべてのホストからのアクセスを許可します。

- 拡張 IP ACL を特定のアプリケーションに適切なコマンドを使用して適用することもできます。存在しない IP ACL への参照は、**permit any** 条件ステートメントと同等です。
- ACNS ソフトウェア リリース 5.4.1 では、IP ACL を使用して、スタンドアロン Content Engine 上で実行している FTP プロキシ サービスへのアクセスを許可または拒否することができます。この機能をサポートするために、次の 2 つの CLI コマンドが ACNS 5.4.1 ソフトウェア リリースに追加されました。

  ```
  ftp-native access-list in {std-acl-num | std-acl-name}
  ftp-native access-list out {ext-acl-num | ext-acl-name}
  ```

  詳細は、「IP ACL による ネイティブ FTP アクセスのコントロール」（p.19-20）を参照してください。

- ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、SNMP と TFTP のアプリケーションには、IP ACL の使用法を設定するための特定の CLI コマンドがあります。これらのコマンドは次のとおりです。

```
snmp-server access-list {std-acl-num | std-acl-name}
tftp-server access-list {std-acl-num | std-acl-name}
```

（注）　**snmp-server access-list** および **tftp-server access-list** グローバル コンフィギュレーション コマンドでは、標準 IP ACL の名前または番号のみが使用できます。拡張 IP ACL の名前や番号は使用できません。

その他のアプリケーション トラフィック（Telnet や SSH など）を制御するには、IP ACL をスタンドアロン Content Engine のインターフェイス（通常、着信トラフィック）に適用します。

- ACNS 5.2.1 ソフトウェアおよびそれ以降のリリースでは、**wccp access-list** グローバル コンフィギュレーション コマンドを使用して、Content Engine が WCCP GRE 着信トラフィックに適用する IP ACL を指定します。

```
wccp access-list {acl-num | acl-name}
```

WCCP アクセス制御リスト機能は、標準と拡張の両アクセス制御リストをサポートし、SNMP および TFTP サーバ アクセス リストの場合と同様に、標準アクセス制御リストだけに制限されることはありません。WCCP アクセス制御リストの詳しい設定方法については、「スタンドアロン Content Engine での WCCP アクセス リストの設定」（p.19-22）を参照してください。

- 標準 IP ACL の場合は、**ip access-list** コマンドの **wildcard** パラメータは常にオプションになります。次の例に示すように、標準 IP ACL で **host** キーワードを指定した場合は、**wildcard** パラメータは使用不可になります。

```
ContentEngine(config)# ip access-list standard 1
ContentEngine(config-std-nacl)# permit ?
  A.B.C.D  Source address
  any      Any source host
  host     A single host address
ContentEngine(config-std-nacl)# permit 10.1.1.1 ?
  A.B.C.D  Source wildcard bits <=== *** Wildcard parameter is optional here ***
<cr>
ContentEngine(config-std-nacl)# permit host 10.1.1.1 ? <=== *** Wildcard parameter
is not allowed here because the host keyword is used***
<cr>
ContentEngine(config-std-nacl)# permit 10.1.1.1
ContentEngine(config-std-nacl)# exit
```

- 拡張 IP ACL の場合は、**host** キーワードを指定しないかぎり、**wildcard** パラメータは必須になります。次の例に示すように、拡張 IP ACL で **host** キーワードを指定した場合は、**wildcard** パラメータは使用不可になります。

```
ContentEngine(config)# ip access-list extended 100
ContentEngine(config-ext-nacl)# permit ?
  <1-255>  An IP Protocol Number
  gre      Cisco's GRE Tunneling
  icmp     Internet Control Message Protocol
  ip       Any IP Protocol
  tcp      Transport Control Protocol
  udp      User Datagram Protocol
ContentEngine(config-ext-nacl)# permit ip ?
  A.B.C.D  Source address
  any      Any source host
  host     A single host address
ContentEngine(config-ext-nacl)# permit ip 10.1.1.1 ?
  A.B.C.D  Source wildcard bits
<=== *** Wildcard parameter is required here because the host keyword is not
specified***
ContentEngine(config-ext-nacl)# permit ip host ?
  A.B.C.D  Source address
ContentEngine(config-ext-nacl)# permit ip host 10.1.1.1 ? <=== *** Wildcard
parameter is not allowed here because the host keyword is used***
  A.B.C.D  Destination address
  any      Any destination host
  host     A single host address
```

- 標準または拡張 IP ACL コンフィギュレーション モードに入っている場合は、編集コマンド（**list**、**delete**、および **move**）を使用して、エントリの表示、特定のエントリ（条件）の削除、またはエントリが評価される順序の変更を行うことができます。

```
ContentEngine(config)# ip access-list standard 1
ContentEngine(config-std-nacl)#?
      delete  Delete a condition
      deny    Specify packets to reject
      exit    Exit from this submode
      insert  Insert a condition
      list    List conditions
      move    Move a condition
      no      Negate a command or set its defaults
      permit  Specify packets to accept
ContentEngine(config-std-nacl)#
```

- **list** コマンドを使用して、条件をマッピングする行番号を特定します。このコマンドは、指定のエントリを表示します（none を指定するとすべてのエントリが表示されます）。このコマンドを使用しない場合、EXEC モードに戻り、**show ip access-list** EXEC コマンドを入力してこのマッピングを取得する必要があります。

  次の例は、**list** コマンドの使用方法を示しています。

```
Content Engine(config-ext-nacl)# list
  1  permit tcp host 10.1.1.1 any
  2  permit tcp host 10.1.1.2 any
  3  permit tcp host 10.1.1.3 any
Content Engine(config-ext-nacl)#
```

- Content Engine のデータベースから IP ACL をすべて削除する方法については、「IP ACL の削除」（p.19-24）を参照してください。

IP ACL コンフィギュレーション モードに関する使用上の注意事項

IP ACL を使用する際は、IP ACL コンフィギュレーション モードに関する次の重要な点に留意してください。

- 標準 IP ACL を操作するには、標準 IP ACL コンフィギュレーション モードに入る必要があります。

```
ContentEngine(config)# ip access-list standard ?
    <1-99>   Standard IP access-list number
    WORD     Access-list name (max 30 characters)
```

- 拡張 IP ACL を操作するには、拡張 IP ACL コンフィギュレーション モードに入る必要があります。

```
ContentEngine(config)# ip access-list extended ?
    <100-199>  Standard IP access-list number
    WORD       Access-list name (max 30 characters)
```

IP ACL の名前に関する使用上の注意事項

IP ACL の名前を作成する際は、次の命名時の注意事項を参照してください。

- IP ACL の名前は、Content Engine 内で一意でなければなりません。
- IP ACL の名前が数値である場合（**ip access-list standard** *acl-num* または **ip access-list extended** *acl-num* など）
  - 数値のみ指定できます（たとえば、101）。
  - 数値 1 ～ 99 は標準 IP ACL を表します。
  - 数値 100 ～ 199 は拡張 IP ACL を表します。
- IP ACL の名前が文字である場合（**ip access-list standard** *acl-name* または **ip access-list extended** *acl-name* など）
  - 名前の先頭は文字でなければなりません（たとえば、snmpaccesslist）。
  - 30 文字まで指定できます。
  - 文字列内に 0 ～ 9 の数字を指定できます（たとえば、snmpaccesslist7）。
  - ほとんどの印刷可能な特殊文字を指定できます。ただし、スペースは指定できません。指定できる特殊文字は、~!@#$%^&*()_+-={}[]\:;'<>,./ です。指定できない特殊文字は、`|"? です。

（注）　スタンドアロン Content Engine で IP ACL を作成または変更する方法については、次の「スタンドアロン Content Engine での IP ACL の作成または変更」の項を参照してください。

スタンドアロン Content Engine での IP ACL の作成または変更

スタンドアロン Content Engine で IP ACL を設定する手順は、次のとおりです。

ステップ 1　グローバル コンフィギュレーション モードで Content Engine CLI にアクセスします。

```
ContentEngine(config)#
```

ステップ 2　グローバル コンフィギュレーション モードから、適切な IP ACL コンフィギュレーション モードにアクセスし、作成、変更、または表示する IP ACL の名前または番号を指定します。

- 標準 IP ACL を作成または変更するには、**ip access-list standard** グローバル コンフィギュレーション コマンドを使用して標準 IP ACL コンフィギュレーション モードに入ります。

  ip access-list standard {*acl-name* | *acl-num*}

  次の例では、59 の ACL 番号を持つ標準 IP ACL の作成または変更方法を示しています。

  ```
  ContentEngine(config)# ip access-list standard 59
  ```

  この CLI は標準 IP ACL コンフィギュレーション モードに入ります。このモードでは、以降のすべてのコマンドが現在の標準 IP ACL に適用され、次のプロンプトが表示されます。

  ```
  ContentEngine(config-std-nacl)#
  ```

- 拡張 IP ACL を作成または変更するには、**ip access-list extended** コマンドを使用して拡張 IP ACL コンフィギュレーション モードに入ります。

  ip access-list extended {*acl-name* | *acl-num*}

  次の例では、test2 という名前の拡張 IP ACL にその名前を指定して作成または変更する方法を示しています。

  ```
  ContentEngine(config)# ip access-list extended test2
  ```

  この CLI は拡張 IP ACL コンフィギュレーション モードに入ります。このモードでは、以降のすべてのコマンドが現在の拡張 IP ACL に適用され、次のプロンプトが表示されます。

  ```
  ContentEngine(config-ext-nacl)#
  ```

ステップ 3　標準 ACL で条件を追加、削除、または変更するには、標準 IP ACL コンフィギュレーション モードから次のコマンドを入力します。

a.　標準 IP ACL に行を追加するには、次の構文を使用します。

たとえば、パケットを通過または廃棄させるかの指定を許可（permit）または拒否（deny）から選択し、その送信元 IP アドレスと送信元 IP ワイルドカード アドレスを入力します。

[**insert** *line-num*] {**deny** | **permit**} {*source-ip* [*wildcard*] | **host** *source-ip* | **any**}

b.　標準 IP ACL から行を削除するには、次の構文を使用します。

delete *line-num*

c.　標準 IP ACL 内で行を別の位置に移動するには、次の構文を使用します。

move *old-line-num new-line-num*

（注）　拡張 IP ACL 条件のリストについては、表 19-4 を参照してください。

ステップ 4　拡張 ACL で条件を追加、削除、または変更するには、拡張 IP ACL コンフィギュレーション モードから次のコマンドを入力します。

a. 拡張 IP ACL から行を削除するには、次の構文を使用します。

```
delete line-num
```

b. 拡張 IP ACL 内で行を別の位置に移動するには、次の構文を使用します。

```
move old-line-num new-line-num
```

c. 拡張 IP ACL に条件を追加するには、選択するプロトコルに従ってオプションを入力します。

- IP の場合は、次の構文を使用して条件を追加します。

```
[insert line-num] {deny | permit} {gre | ip | proto-num}
{source-ip wildcard | host source-ip | any} {dest-ip wildcard |
host dest-ip | any}

[no] {deny | permit} {gre | ip | proto-num} {source-ip wildcard |
host source-ip | any} {dest-ip wildcard | host dest-ip | any}
```

- TCP の場合は、次の構文を使用して条件を追加します。

```
[insert line-num] {deny | permit} tcp {source-ip wildcard |
host source-ip | any} [operator port [port]] {dest-ip wildcard |
host dest-ip | any} [operator port [port]] [established]

no {deny | permit} tcp {source-ip wildcard | host source-ip | any}
[operator port [port]] {dest-ip wildcard | host dest-ip | any}
[operator port [port]] [established]
```

- UDP の場合は、次の構文を使用して条件を追加します。

```
[insert line-num] {deny | permit} udp {source-ip wildcard |
host source-ip | any} [operator port [port]] {dest-ip wildcard |
host dest-ip | any} [operator port [port]]

no {deny | permit} udp {source-ip wildcard | host source-ip | any}
[operator port [port]] {dest-ip wildcard | host dest-ip | any} |
[operator port [port]]
```

- ICMP の場合は、次の構文を使用して条件を追加します。

```
[insert line-num] {deny | permit} icmp {source-ip wildcard |
host source-ip | any} {dest-ip wildcard | host dest-ip | any}
[icmp-type [code] | icmp-msg]

no {deny | permit} icmp {source-ip wildcard | host source-ip | any}
{dest-ip wildcard | host dest-ip | any} [icmp-type [code] | icmp-msg]
```

（注）　拡張 IP ACL では、**host** キーワードを指定しない場合は、**wildcard** パラメータが必要になります。特定の ICMP メッセージ タイプやコードとの照合に使用できるキーワードのリストについては、表 19-3 を参照してください。サポートされている UDP と TCP キーワードのリストについては、表 19-1 と表 19-2 を参照してください。拡張 IP ACL 条件のリストについては、表 19-5 を参照してください。

ステップ 5　標準 IP ACL に別の条件を追加するには、ステップ 3 を繰り返し実行してください。拡張 IP ACL に別の条件（エントリ）を追加するには、ステップ 4 を繰り返し実行してください。

ステップ 6　**interface** と **ip access-group** の各コマンドを使用して、この IP ACL をアクティブにして Content Engine の特定のインターフェイスに適用します。

IP ACL をアクティブにして、特定のインターフェイスに適用する方法については、「インターフェイスでの IP ACL のアクティブ化」（p.19-17）と「アプリケーションへの IP ACL の適用」（p.19-18）を参照してください。

拡張 IP ACL のキーワードリスト

表 19-1 に、拡張 IP ACL で使用可能な UDP キーワードを表示します。

表 19-1　UDP キーワードとポート番号

CLI キーワード	説明	UDP ポート番号
bootpc	BOOTP クライアント	68
bootps	BOOTP サーバ	67
domain	Domain Name System（DNS; ドメイン ネーム システム）	53
mms	MMS プロトコル	1755
netbios-dgm	NetBIOS データグラム サービス	138
netbios-ns	NetBIOS ネーム サービス	137
netbios-ss	NetBIOS セッション サービス	139
nfs	NFS サービス	2049
ntp	Network Time Protocol（NTP）	123
snmp	SNMP（簡易ネットワーク管理プロトコル）	161
snmptrap	SNMP トラップ	162
tacacs	TACACS	49
tftp	Trivial File Transfer Protocol（TFTP; 簡易ファイル転送プロトコル）	69
wccp	Cisco Web Cache Communication Protocol（WCCP）	2048

表 19-2 に、拡張 IP ACL で使用可能な TCP キーワードを表示します。

表 19-2　TCP キーワードとポート番号

CLI キーワード	説明	TCP ポート番号
domain	DNS	53
exec	Exec（RCP）	512
ftp	FTP（ファイル転送プロトコル）	21
ftp-data	FTP データ接続（ほとんど使用されない）	20
https	Secure HTTP	443
nfs	NFS サービス	2049
rtsp	Real-Time Streaming Protocol（RTSP）	554
ssh	SSH ログイン	22
tacacs	TACACS	49
telnet	Telnet	23
www	World Wide Web（HTTP）	80

表 19-3 に、特定の ICMP メッセージ タイプとコードの照合に使用できるキーワードを示します。

表 19-3　ICMP メッセージ タイプとコードのキーワード

administratively-prohibited	**alternate-address**
conversion-error	**dod-host-prohibited**
dod-net-prohibited	**echo**
echo-reply	**general-parameter-problem**
host-isolated	**host-precedence-unreachable**
host-redirect	**host-tos-redirect**
host-tos-unreachable	**host-unknown**
host-unreachable	**information-reply**
information-request	**mask-reply**
mask-request	**mobile-redirect**
net-redirect	**net-tos-redirect**
net-tos-unreachable	**net-unreachable**
network-unknown	**no-room-for-option**
option-missing	**packet-too-big**
parameter-problem	**port-unreachable**
precedence-unreachable	**protocol-unreachable**
reassembly-timeout	**redirect**
router-advertisement	**router-solicitation**
source-quench	**source-route-failed**
time-exceeded	**timestamp-reply**
timestamp-request	**traceroute**
ttl-exceeded	**unreachable**

IP ACL の条件

表 19-4 で、標準 IP ACL の条件を説明します。

表 19-4　標準 IP ACL の条件

パラメータ	説明
insert	（任意）標準 IP ACL 内の指定の行番号の直後に条件を挿入します。
line-num	標準 IP ACL 内の特定の行番号のエントリを指定します。
deny	指定の条件と一致するパケットを廃棄します。
permit	指定の条件と一致するパケットを許可し、以降の処理を行います。
source-ip	送信元 IP アドレスパケットの送信元のネットワークまたはホストの番号。この番号は、32 ビットの 4 つに区切られたドット付き 10 進数表記で指定します（たとえば、0.0.0.0）。
wildcard	先行の IP アドレスの照合部分（4 桁のドット区切り表記）を指定します。照合するビットは、0 の数値で識別されます。無視するビットは 1 で識別されます。 標準 IP ACL の場合は、**wildcard** parameter of the **ip access-list** コマンドは常にオプションになります。標準 IP ACL で **host** キーワードを指定した場合は、**wildcard** パラメータは使用不可になります。
host	直後の IP アドレスと照合します。
any	任意の IP アドレスと照合します。

表 19-4　標準 IP ACL の条件（続き）

パラメータ	説明
delete	指定のエントリ（条件）を標準 IP ACL から削除します。
line-num	標準 IP ACL 内の特定の行番号のエントリを指定します。
list	指定のエントリを表示します（none を指定するとすべてのエントリが表示されます）。
start-line-num	（任意）リストの開始行番号
end-line-num	（任意）リストの最終行番号
move	標準 IP ACL に指定されているエントリをリスト内の別の位置に移動します。
old-line-num	移動するエントリの行番号を指定します。
new-line-num	エントリの新しい位置を指定します。既存のエントリが標準 IP ACL 内でこの新しい位置に移動されます。

表 19-5 で、拡張 IP ACL の条件を説明します。

表 19-5　拡張 IP ACL の条件

パラメータ	説明
insert	（任意）条件を拡張 IP ACL 内の指定の行番号に挿入します。
line-num	拡張 IP ACL 内の特定の行番号のエントリを指定します。
deny	指定の条件と一致するパケットを廃棄します。
permit	指定の条件と一致するパケットを許可し、以降の処理を行います。
source-ip	送信元 IP アドレス
wildcard	先行の IP アドレスの照合部分（4 桁のドット区切り表記）を指定します。照合するビットは、0 の数値で識別されます。無視するビットは 1 で識別されます。 拡張 IP ACL の場合は、**host** キーワードを指定しないかぎり、**wildcard** パラメータは必須になります。拡張 IP ACL で **host** キーワードを指定した場合は、**wildcard** パラメータは使用不可になります。
host	直後の IP アドレスと照合します。
any	任意の IP アドレスと照合します。
gre	Generic Routing Encapsulation（GRE; 総称ルーティング カプセル化）プロトコルを使用したパケットを照合します。
ip	すべての IP パケットを照合します。
proto-num	IP プロトコル番号を指定します。
tcp	TCP プロトコルを使用したパケットを照合します。
udp	UDP プロトコルを使用したパケットを照合します。

operator	（任意）指定のポートで使用する演算子を指定します。演算子は、**lt**= より小さい、**gt**= より大きい、**eq** = 等しい、**neq**= 等しくない、**range**= 範囲、となります。次の例では、拡張 IP ACL で **eq** 演算子を使用しています。
	
	`ContentEngine(config)# ip access-list extended example`
	`ContentEngine(config-ext-nacl)# permit tcp any any eq www`
	`ContentEngine(config-ext-nacl)# permit tcp host 10.1.1.5 any eq ssh`

表 19-5　拡張 IP ACL の条件（続き）

パラメータ	説明

port	（任意）番号（0 ～ 65535）またはキーワードを使用してポートを指定します。その場合、2 つのポート番号を **range** 演算子で指定する必要があります。TCP では、**domain**、**exec**、**ftp**、**ftp-data**、**https**、**mms**、**nfs**、**rtsp**、**ssh**、**tacacs**、**telnet**、**www** のいずれかを使用します。UDP では、**bootpc**、**bootps**、**domain**、**mms**、**netbios-dgm**、**netbios-ns**、**netbios-ss**、**nfs**、**ntp**、**snmp**、**snmptrap**, **tacacs**, **tftp**、**wccp** のいずれかを使用します。次に例を示します。
	`ContentEngine(config)# ip access-list extended example`
	`ContentEngine(config-ext-nacl)# permit tcp any any eq www`
	`ContentEngine(config-ext-nacl)# permit tcp host 10.1.1.5 any eq ssh`

dest-ip	送信先 IP アドレス
established	（任意）TCP パケットを ACK または RST ビット セットと照合します。
icmp	ICMP パケットを照合します。
icmp-type	（任意）番号（0 ～ 255）で表される CMP メッセージ タイプで照合します。
code	（任意）*icmp-type* と一緒に使用して、0 ～ 255 の番号で表される ICMP コード タイプでさらに照合します。
icmp-msg	（任意）表 19-3 に記載のキーワードで表される ICMP メッセージ タイプとコード タイプの組み合わせで照合します。
delete	拡張 IP ACL から指定のエントリ（条件）を削除します。
line-num	拡張 IP ACL 内の特定の行番号のエントリを指定します。
list	指定のエントリを表示します（none を指定するとすべてのエントリが表示されます）。
start-line-num	（任意）リストの開始行番号
end-line-num	（任意）リストの最終行番号
move	リスト内の指定のエントリをリスト内の別の位置に移動します。
old-line-num	移動するエントリの行番号を指定します。
new-line-num	エントリの新しい位置を指定します。既存のエントリがアクセス リスト内のこの位置に移動されます。
exit	CLI グローバル コンフィギュレーション モードのプロンプトに戻ります。

インターフェイスでの IP ACL のアクティブ化

ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、さまざまなサービスを特定のインターフェイスに関連付けることができます。スタンドアロン Content Engine の特定のインターフェイスで IP ACL をアクティブにするには、**ip access-group** インターフェイス コンフィギュレーション コマンドを使用します。各インターフェイスで 1 つの発信 IP ACL と 1 つの着信 IP ACL を使用できます。

ip access-group コマンドを入力する前に、IP ACL を適用するインターフェイスのインターフェイス コンフィギュレーション モードに入ります。

次のコマンドを使用すると、acl-out という名前の IP ACL をインターフェイス FastEthernet スロット 0/ ポート 0 上の発信トラフィックに適用し、アクティブにできます。

```
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)# ip access-group acl-out out
```

次のコマンドを使用すると、example という名前の IP ACL をインターフェイス GigabitEthernet ポート 1/ スロット 0 上の着信トラフィックに適用し、アクティブにできます。

```
ContentEngine(config)# interface gigabitethernet 1/0
ContentEngine(config-if)# ip access-group example in
ContentEngine(config-if)# exit
```

スタンドアロン Content Engine の特定のインターフェイスに IP ACL を適用し、アクティブにする手順は、次のとおりです。

ステップ 1　IP ACL を適用するインターフェイスのインターフェイス コンフィギュレーション モードに入ります。

たとえば、次の例では、Content Engine のインターフェイス FastEthernet スロット 0/ ポート 0 に対してインターフェイス コンフィギュレーション モードに入る方法を示しています。

```
ContentEngine(config)# interface FastEthernet 0/0
ContentEngine(config-if)#
```

ステップ 2　指定したインターフェイスに定義済み IP ACL を適用します。

たとえば、次の例では、acl-out という名前の定義済み IP ACL をインターフェイス FastEthernet スロット 0/ ポート 0 上の発信トラフィックに適用する方法を示しています。

```
ContentEngine(config-if)# ip access-group acl-out out
```

表 19-6 に、**ip access-group** コマンドのパラメータの説明を示します。

表 19-6　ip access-group CLI コマンドのパラメータ

パラメータ	説明
acl-name	英数字の ID を 30 文字まで指定できます。先頭には、現在のインターフェイスに適用する IP ACL を識別するための文字を指定する必要があります。
acl-num	現在のインターフェイスに適用する IP ACL を識別する数値の識別子（1 ～ 99 は標準 IP ACL を表し、100 ～ 199 は拡張 IP ACL を表します）。
in	指定の IP ACL を現在のインターフェイス上の着信パケットに適用します。
out	指定の IP ACL を現在のインターフェイス上の発信パケットに適用します。

アプリケーションへの IP ACL の適用

ACNS 5.4.1 ソフトウェアおよびそれ以降のリリースでは、FTP には、FTP ネイティブ プロキシ モードおよび透過的にリダイレクトされた（WCCP でリダイレクトされた）接続に対する IP ACL の使用法を設定するための次の特定の CLI コンフィギュレーション コマンドがあります。

ftp-native access-list in {*std-acl-num* | *std-acl-name*}

ftp-native access-list out {*ext-acl-num* | *ext -acl-name*}

着信 FTP ネイティブ接続のタイプ（プロキシ モード接続および透過的にリダイレクトされた［WCCP でリダイレクトされた］接続）の標準 ACL を設定するには、**ftp-native access-list in** グローバル コンフィギュレーション コマンドを使用します。標準 ACL の場合、アクセス制御は、ネイティブ FTP 要求を送信した FTP クライアントの送信元アドレスに基づきます。

発信 FTP ネイティブ接続のタイプ（プロキシ モード接続および透過的にリダイレクトされた［WCCP でリダイレクトされた］接続）の拡張 ACL を設定するには、**ftp-native access-list out** グローバル コンフィギュレーション コマンドを使用します。拡張 ACL の場合、アクセス制御は、送信元アドレスと宛先アドレスに基づきます。

ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、SNMP と TFTP には、IP ACL の使用法を設定するための特定の CLI コンフィギュレーション コマンドがあります。

snmp-server access-list {*std-acl-num* | *std-acl-name*}

tftp-server access-list {*std-acl-num* | *std-acl-name*}

（注）　**snmp-server access-list** および **tftp-server access-list** グローバル コンフィギュレーション コマンドでは、標準 IP ACL の名前または番号のみが使用できます。拡張 IP ACL の名前や番号は使用できません。

ACNS 5.2.1 ソフトウェアおよびそれ以降のリリースでは、**wccp access-list** グローバル コンフィギュレーション コマンドを使用して、Content Engine が自身で受信する、カプセル化された WCCP GRE 着信トラフィックに適用する IP アクセス リストを指定します。

wccp access-list {*acl-num* | *acl-name*}

WCCP アクセス リスト機能は、標準と拡張の両アクセス リストをサポートし、SNMP および TFTP サーバ アクセス リストの場合と同様に、標準アクセス リストだけに制限されることはありません。

その他のアプリケーション トラフィック（Telnet や SSH など）を制御するには、IP ACL をスタンドアロン Content Engine のインターフェイス（通常、着信トラフィック）に適用します。

（注）　ACNS 5.1 ソフトウェアおよびそれ以降のリリースでは、TFTP プロトコルを使用したアクセスをユーザに対して許可または拒否するために、**ip access-list** グローバル コンフィギュレーション コマンドを使用する必要があります。IP ACL を TFTP サービスに設定しないかぎり、コンテンツのセキュリティが危険にさらされ、TFTP は正しく機能しません。

ACNS 5.0 ソフトウェアでは、デフォルト設定により、TFTP アクセスがユーザに拒否されていました。ユーザに対してアクセスを許可するために、管理者は **trusted-host** コマンドを使用する必要がありました。ACNS 5.1 ソフトウェアでは、**trusted-host** コマンドの使用は推奨されていません。そのため、リリース 5.1 より前の ACNS ソフトウェア リリースを実行している Content Engine で **trusted-host** コマンドを使用し、デバイスを ACNS ソフトウェア 5.1 およびそれ以降のリリースにアップグレードすると、**trusted-host** コマンドは CLI に表示されますが、TFTP プロトコルにはまったく作用しません。信頼できるホストの設定を削除するには、**no trusted-host** コマンドを使用します。

IP ACL による SNMP アクセスのコントロール

標準 IP ACL を使用して、スタンドアロン Content Engine の SNMP エージェントへのアクセスをコントロールする手順は、次のとおりです。

ステップ 1　**ip access-list standard** コマンドを使用して、Content Engine の SNMP エージェントへのアクセスをコントロールするための IP ACL を作成します。

ステップ 2　この IP ACL を SNMP サーバ（スタンドアロン Content Engine）に関連付け、Content Engine でこの標準 IP ACL をアクティブにします。

```
ContentEngine(config)# snmp-server access-list {std-acl-num | std-acl-name}
```

- *std-acl-name* は、この Content Engine に関連付ける標準 IP ACL の名前です。
- *std-acl-num* は、この Content Engine に関連付ける標準 IP ACL の番号です。

Content Engine の SNMP エージェントは、着信パケットを許可または廃棄する前に、指定の IP ACL（たとえば、ACL 1）と照合します。

```
ContentEngine(config)# snmp-server access-list 1
```

IP ACL による TFTP アクセスのコントロール

標準 IP ACL を使用して、スタンドアロン Content Engine の TFTP サービスへのアクセスをコントロールする手順は、次のとおりです。

ステップ 1　スタンドアロン Content Engine で実行中の TFTP サービスへのアクセスをコントロールするためのアクセス リストを作成します。

たとえば、次のコマンドを使用すると、192.168.1.0 サブネットワーク上の TFTP クライアントに対して、TFTP サービスへのアクセスを許可するアクセス リストを定義できます。

```
ContentEngine(config)# ip access-list standard 2
ContentEngine(config-std-nacl)# permit 192.168.1.0 0.0.0.255
ContentEngine(config-std-nacl)# exit
ContentEngine(config)#
```

ステップ 2　この IP ACL（ACL 2）を TFTP サーバ（スタンドアロン Content Engine）に関連付け、Content Engine でこの標準 IP ACL をアクティブにします。Content Engine は、実行中の TFTP サービスへのアクセスを許可または拒否する前に、指定されているアクセス制御リストと照合します。

次の例では、Content Engine はアクセス制御リスト 2 と照合してから、TFTP アクセスをユーザ（TFTP クライアント）に対して許可または拒否します。

```
ContentEngine(config)# tftp-server access-list 2
```

IP ACL による ネイティブ FTP アクセスのコントロール

ACNS 5.4.1 ソフトウェアおよびそれ以降のリリースでは、スタンドアロン Content Engine で実行中のネイティブ FTP プロキシ サービスへのアクセスを許可または拒否する IP ACL を使用できます。次の着信ネイティブ FTP 要求のタイプでサポートされます。

- 非透過 FTP ネイティブ要求
- 透過 FTP ネイティブ要求（WCCP 対応のルータによって Content Engine に透過的に代行受信され、リダイレクトされる着信 FTP ネイティブ要求）

この新機能をサポートするために、次の 2 つの CLI コマンドが ACNS 5.4.1 ソフトウェア リリースに追加されました。

ftp-native access-list in {*std-acl-num* | *std-acl-name*}
ftp-native access-list out {*ext-acl-num* | *ext-acl-name*}

標準 IP ACL の場合、着信 FTP ネイティブ要求のアクセス制御は、送信元アドレスに基づいています。拡張 IP ACL の場合、オリジン サーバに発信される FTP ネイティブ要求のアクセス制御は、送信元アドレスと宛先アドレスに基づいています。

スタンドアロン Content Engine で実行中のネイティブ FTP プロキシ サービスへのアクセスをコントロールする IP ACL を使用する手順は、次のとおりです。

ステップ 1　スタンドアロン Content Engine で実行中のネイティブ FTP プロキシ サービスへのアクセスをコントロールするためのアクセス リストを作成します。

たとえば、次のコマンドを使用すると、192.168.1.0 サブネットワーク上の FTP クライアントに対してネイティブ FTP プロキシ サービスへのアクセスを許可するアクセス リストを定義できます。

```
ContentEngine(config)# ip access-list standard 3
ContentEngine(config-std-nacl)# permit 192.168.1.0 0.0.0.255
ContentEngine(config-std-nac1)# exit
ContentEngine(config)#
```

ステップ 2　この IP ACL（ACL 3）をネイティブ FTP プロキシ サービスに関連付け、Content Engine でこの標準 IP ACL をアクティブにします。Content Engine は、ネイティブ FTP プロキシ サービスへのアクセスを許可または拒否する前に、指定されているアクセス制御リストと照合します。

次の例では、アクセス制御リスト 3 と照合してから、FTP プロキシ アクセスをユーザ（たとえば、Reflection X クライアント、WS-FTP クライアント、あるいは UNIX または DOS コマンドライン FTP プログラムなどの FTP クライアント）に対して許可または拒否するように Content Engine を設定します。

```
ContentEngine(config)# ftp-native access-list in 3
```

ステップ 3　（任意）Content Engine が着信プロキシ モード接続に対する応答で FTP クライアントに送信する、カスタマイズされた応答メッセージを設定します。

ACNS 5.4.1 ソフトウェアおよびそれ以降のリリースでは、**ftp-native custom-message** グローバル コンフィギュレーション コマンドを使用して、Content Engine が着信プロキシ モード接続に対する応答で FTP クライアントに送信する、カスタマイズされた応答メッセージを設定できます。**ftp-native custom-message** EXEC コマンドを使用して、次のいずれかのカスタム メッセージを含むファイルを作成、アップロード、およびダウンロードできます。

- FTP クライアントからのプロキシ モード接続を受け入れるカスタム ウェルカム メッセージ

- ネイティブ FTP プロキシ サービスに設定された IP ACL に基づいたネイティブ FTP プロキシ サービスへのアクセスをユーザ（FTP クライアント）に対して拒否するカスタム エラー メッセージ

このトピックに関する詳細は、「FTP ネイティブ要求に対する FTP プロキシ応答用のカスタム メッセージの作成」（p.5-21）を参照してください。

ステップ 4　ネイティブ FTP 要求を FTP クライアントから Content Engine に発信し、定義されたアクセス制御リストが正常に機能していることを確認します。

次の例では、192.168.1.0 サブネットワーク上の FTP クライアントからネイティブ FTP 要求を発信します。この場合、FTP クライアントは UNIX コマンドライン FTP プログラムで、Content Engine の IP アドレスは 10.1.1.50 です。送信元アドレス（FTP クライアントの IP アドレス）が 192.168.1.0 サブネットワーク上にあるため、この FTP クライアントの FTP サービスに対するアクセスが許可され、Content Engine が FTP クライアントに対してウェルカム メッセージを表示します。

```
shell# ftp ñd 10.1.1.50 8501
Connected to 10.1.1.50
220 Welcome to FTP-proxy.  Login to the proxy using username and password.
Name (10.1.1.50:admin):
```

IP ACL による WCCP アクセスのコントロール

標準または拡張 IP ACL を使用して、スタンドアロン Content Engine の WCCP サービスへのアクセスをコントロールする手順は、次のとおりです。

ステップ 1　Content Engine での WCCP アクセスをコントロールするための標準または拡張 IP ACL を作成します。

a.　標準 IP ACL を作成するには、**ip access-list standard** コマンドを使用します。

b.　拡張 IP ACL を作成するには、**ip access-list extended** コマンドを使用します。

ステップ 2　IP ACL を Content Engine に関連付け、Content Engine でこの IP ACL をアクティブにします。

```
ContentEngine(config)# wccp access-list {acl-num | acl-name}
```

- *acl-name* は、この Content Engine に関連付ける標準または拡張 IP ACL の名前です。
- *acl- num* は、この Content Engine に関連付ける標準または拡張 IP ACL の番号です。

Content Engine は、指定の IP ACL（たとえば、ACL 2）を WCCP GRE 着信トラフィックに適用します。

```
ContentEngine(config)# wccp access-list 2
```

スタンドアロン Content Engine での WCCP アクセス リストの設定

ACNS 5.2.1 ソフトウェアおよびそれ以降のリリースでは、**wccp access-list** グローバル コンフィギュレーション コマンドを使用して、Content Engine が自身で受信する、カプセル化された WCCP GRE 着信トラフィックに適用する IP アクセス リストを指定します。

wccp access-list {*acl-name* | *acl-number*}

acl-name または *acl-number* は、標準または拡張 IP アクセス リストです。

デフォルトでは、いずれの WCCP アクセス リストも設定されません。そのため、WCCP アクセス リストは、Content Engine の設定の一部として表示されません。

次に示すのは、Content Engine に WCCP アクセス リストが設定されている場合の **show ip access-list** EXEC コマンドの出力例です。

```
Content Engine# show ip access-list
Space available:
    48 access lists
   497 access list conditions

Standard IP access list test
   1 permit 10.1.1.1
     (implicit deny any:0 matches)
   total invocations:0
Extended IP access list no_www.linux.org
   1 deny tcp any host 10.1.1.1 (29 matches)
   2 permit ip any any (30 matches)
     (implicit fragment permit:0 matches)
     (implicit deny ip any any:0 matches)
   total invocations:59

Interface access list references:
  GigabitEthernet 2/0  inbound   pc_test (Not Defined)

Application access list references:
  snmp-server                    standard  test
    UDP ports:none
  tftp_server                    standard  test4
    UDP ports:   69 (List Not Defined)
  WCCP                           either    no_www.linux.org
    Any IP Protocol
Content Engine#
```

ACNS 5.2.1 ソフトウェアおよびそれ以降のリリースでは、**show wccp gre** EXEC コマンドの出力には、WCCP アクセス リスト機能に関連する 2 つのカウンタが含まれています。

```
---------------
Packets w/WCCP GRE received too small:     0
Packets dropped due to IP access-list deny:29
---------------
```

最初のカウンタは、サイズが小さすぎて IP パケット ヘッダー全体が収まらないために廃棄された、正しくカプセル化された WCCP GRE パケットの数を表します。

2 番めのカウンタは、指定の WCCP アクセス リストによって拒否されたために廃棄されたパケットの数を表します。

次に示すのは、Content Engine で WCCP アクセス リストが定義されている場合の **show wccp gre** EXEC コマンドからの出力例です。

```
Content Engine# show wccp gre
Transparent GRE packets received:          366
Transparent non-GRE packets received:      0
Transparent non-GRE packets passed through:0
Total packets accepted:                    337
Invalid packets received:                  0
Packets received with invalid service:     0
Packets received on a disabled service:    0
Packets received too small:                0
Packets dropped due to zero TTL:           0
Packets dropped due to bad buckets:        0
Packets dropped due to no redirect address:0
Connections bypassed due to load:          0
Packets sent back to router:               0
Packets sent to another CE:                0
GRE fragments redirected:                  0
Packets failed GRE encapsulation:          0
Packets dropped due to invalid fwd method: 0
Packets dropped due to insufficient memory:0
Packets bypassed, no conn at all:          0
Packets bypassed, no pending connection:   0
Packets due to clean wccp shutdown:        0
Packets bypassed due to bypass-list lookup:0
Packets received with client IP addresses: 0
Conditionally Accepted connections:        0
Conditionally Bypassed connections:        0
Packets w/WCCP GRE received too small:     0
Packets dropped due to IP access-list deny:29
Content Engine#
```

（注）　前述の情報は、**show statistics wccp gre** EXEC コマンドを入力して出力することもできます。

設定例

次の例では、**ip access-list extended** グローバル コンフィギュレーション コマンドを使用して、*example* という名前の拡張 IP ACL を作成する方法を示しています。この拡張 IP ACL は、すべての Web トラフィックを許可する一方、SSH を使用した特定のホスト（ホスト 10.1.1.5）の管理アクセスのみを許可します。

```
ContentEngine(config)# ip access-list extended example
ContentEngine(config-ext-nacl)# permit tcp any any eq www
ContentEngine(config-ext-nacl)# permit tcp host 10.1.1.5 any eq ssh
ContentEngine(config-ext-nacl)# exit
```

次に示すのは、インターフェイス アクセス リストとアプリケーション アクセス リストを使用するように設定したスタンドアロン Content Engine の例です。

```
ContentEngine# show ip access-list
Space available:
    47 access lists
   492 access list conditions

Standard IP access list 1
  1 permit 10.1.1.2
  2 deny   10.1.2.1
    (implicit deny any: 2 matches)
  total invocations: 2
Extended IP access list 100
  1 permit tcp host 10.1.1.1 any
  2 permit tcp host 10.1.1.2 any
  3 permit tcp host 10.1.1.3 any
    (implicit fragment permit: 0 matches)
    (implicit deny ip any any: 0 matches)
  total invocations: 0
Standard IP access list test
  1 permit 1.1.1.1 (10 matches)
  2 permit 1.1.1.3
  3 permit 1.1.1.2
    (implicit deny: 2 matches)
  total invocations: 12
Interface access list references:
  FastEthernet    0/0  inbound   100
Application access list references:
  tftp_server                     standard  1
    UDP ports:    69
```

IP ACL の削除

Content Engine のデータベースから IP ACL とネットワーク インターフェイスとアプリケーション内のすべての条件と参照を削除する手順は、次のとおりです。

ステップ 1　グローバル コンフィギュレーション モードで Content Engine CLI にアクセスします。

```
ContentEngine(config)#
```

ステップ 2　削除する IP ACL の名前または番号を指定します。

- 標準 IP ACL を削除するには、削除する標準 IP ACL を指定します。

  ```
  ContentEngine(config)# no ip access-list standard {acl-name | acl-num}
  ```

  次の例は、test2 という名前の標準 IP ACL を削除する方法を示しています。

  ```
  ContentEngine(config)# no ip access-list standard test2
  ```

- 拡張 IP ACL を削除するには、削除する拡張 IP ACL を指定します。

  ```
  ContentEngine(config)# no ip access-list extended {acl-name | acl-num}
  ```

  次の例は、example という名前の拡張 IP ACL を削除する方法を示しています。

  ```
  ContentEngine(config)# no ip access-list extended example
  ```

IP ACL 設定の表示

Content Engine で現在定義されている IP ACL の設定を表示するには、**show ip access-list** EXEC コマンドを使用します。

show ip access-list [*acl-name* | *acl-num*]

show ip access-list EXEC コマンドを使用して、現在のシステム（この場合は、スタンドアロン Content Engine）に定義されている IP ACL に関する設定情報を表示できます。特定の IP ACL を名前または番号で指定しないかぎり、定義済みのすべての IP ACL に関する情報が次の項目を含めて表示されます。

- 新規のリストと条件に使用可能なスペース
- 定義済みの IP ACL
- インターフェイスとアプリケーションからの参照

次の例は、特定の IP ACL を指定しなかった場合の **show ip access-list** EXEC コマンドの出力例です。

```
ContentEngine# show ip access-list
Space available:
    47 access lists
   492 access list conditions

Standard IP access list 1
  1 permit 10.1.1.2
  2 deny   10.1.2.1
    (implicit deny any: 2 matches)
  total invocations: 2
Extended IP access list 100
  1 permit tcp host 10.1.1.1 any
  2 permit tcp host 10.1.1.2 any
  3 permit tcp host 10.1.1.3 any
    (implicit fragment permit: 0 matches)
    (implicit deny ip any any: 0 matches)
  total invocations: 0
Standard IP access list test
  1 permit 1.1.1.1 (10 matches)
  2 permit 1.1.1.3
  3 permit 1.1.1.2
    (implicit deny: 2 matches)
  total invocations: 12

Interface access list references:
  FastEthernet    0/0  inbound   100
Application access list references:
  tftp_server                     standard  1
    UDP ports:    69
```

次の例は、test という名前の IP ACL に対する **show ip access-list** EXEC コマンドの出力例です。

```
ContentEngine# show ip access-list test
Standard IP access list test
  1 permit 1.1.1.1 (10 matches)
  2 permit 1.1.1.3
  3 permit 1.1.1.2
    (implicit deny: 2 matches)
  total invocations: 12
```

（注） パケットの数が 0 より大きい場合に限り、条件ステートメントに一致したパケットの数が表示されます。

IP ACL カウンタのクリア

Content Engine で IP ACL をクリアし、IP ACL の統計情報をリセットするには、**clear ip access-list counter** EXEC コマンドを使用します。

```
ContentEngine# clear ip access-list counters {acl-name | acl-num}
```

この EXEC コマンドを使用して、既存のすべての IP ACL の条件ステートメントに関連した IP ACL カウンタをクリアします。IP ACL の名前または番号を指定すると、指定されたリストのカウンタのみがクリアされます。