CSS のアクセス制御

この章では、ネットワーク トラフィックなど CSS へのアクセスの設定方法を説明します。この章の記載情報は、特に指示がない限り、CSS の全モデルに共通です。

この章の主な内容は次のとおりです。

管理者のユーザ名とパスワードの変更

CSS に初めてログインするときは、デフォルトのユーザ名 **admin** とデフォルトのパスワード **system** を小文字で入力します。セキュリティを確保するため、管理者のユーザ名とパスワードは変更する必要があります。出荷時には、すべての CSS で管理者のユーザ名とパスワードが同一に設定されているため、CSS のセキュリティが損なわれる可能性があります。

管理者のユーザ名とパスワードは、nonvolatile random access memory（NVRAM; 不揮発性 RAM）に保持されています。CSS を再度ブートするたびに、ユーザ名とパスワードが NVRAM から読み取られ、ユーザ データベースに書き込まれます。管理者のユーザ名には、デフォルトで SuperUser ステータスが割り当てられています。

管理者のユーザ名とパスワードは変更できますが、これらの値は NVRAM 内に保持されているため、完全に削除することはできません。管理者のユーザ名を **no username** コマンドで削除すると、そのユーザ名は running-config ファイルから削除されますが、再度ブートすると NVRAM から復元されます。

管理者のユーザ名とパスワードを変更するには、**username-offdm** *name* **password** *text* コマンドを使用します。

（注） ブート時に **Offline DM** メニューの **Security Options** メニューを使用して、管理者のユーザ名とパスワードを変更することもできます。**Offline DM** メニューの詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

たとえば、デフォルトの管理者のユーザ名とパスワードを変更するには、次のように入力します。

```
(config)# username-offdm bobo password secret
```

ユーザ名とパスワードの作成

CSS にログインするには、ユーザ名とパスワードが必要です。CSS は、管理者や技術者用のユーザ名を含め、最大で 32 個のユーザ名をサポートします。各ユーザには、SuperUser か User のステータスを割り当てることができます。

- **User** ：一部のコマンド群を使って CSS パラメータの監視や表示を実行できるが、CSS パラメータを変更することはできない。User ステータスのプロンプトには、末尾に **>** が付きます。
- **SuperUser** :User ステータスで使用できる各コマンドを含む CSS のすべての CLI コマンドを使って CSS を設定できる。SuperUser ステータスのプロンプトには、末尾に **#** が付きます。

  SuperUser モードでは、グローバル設定モードと、その下位の各設定モードを利用できます。新しいユーザを設定する際に **superuser** オプションを指定しないと、新しいユーザはデフォルトで User ステータスになります。

注意　ユーザ名やパスワードを作成したり変更したりできるのは、管理者または技術者として認識される CSS ユーザだけです。この制限は、**restrict user-database** コマンドが実行済みかどうかによって左右されます。

CSS にログインするためのユーザ名とパスワードは、**username** コマンドで作成します。このグローバル設定モードのコマンドのシンタックスは次のとおりです。

username *name* [**des-password**|**password**] *password* {**superuser**} {**dir-access** *access*}

次の例では、ユーザ名 *picard*、パスワード *captain* のユーザが、SuperUser ステータスで作成されます。

```
(config)# username picard password "captain" superuser
```

このコマンドのオプションと変数は次のとおりです。

- *name* ：割り当てまたは変更するユーザ名を設定する。スペースを含まない 16 文字以内の文字列を、引用符で囲まずに指定します。既存のユーザ名のリストを表示するには、**username ?** コマンドを使用します。

- **des-password** ：Data Encryption Standard （DES; データ暗号化規格）でパスワードを暗号化する。このオプションは、スクリプトや起動設定ファイルとして使用するファイルを作成する場合のみに使用します。DES のパスワードとして、6 ～ 64 文字の文字列を引用符で囲まずに、大文字と小文字を区別して入力します。
- **password** ：パスワードを暗号化しない。このオプションは、コマンド行で必要に応じてユーザを作成するときに使用します。
- *password* ：パスワード。スペースを含まない 6 ～ 16 文字の文字列を、引用符で囲まずに指定します。CSS では、パーセント記号（%）を除いたすべての特殊文字をパスワードに使用できます。

（注） **des-password** オプションを指定すると、CSS に正しくログインするには、暗号化されたパスワードが必要になります。CSS 暗号化パスワードは実行設定に含まれています。CSS　の実行設定を表示するには、**show running-config** コマンドを使用します（「ユーザ名とパスワードの作成」参照）。

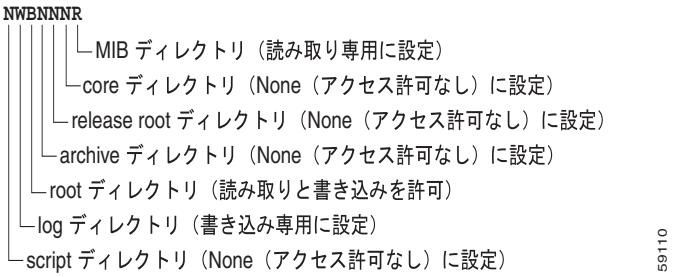
- **superuser** ：ユーザに SuperUser モードの利用を許可する。このオプションを指定しない場合、ユーザが利用できるのは User モードだけです。
- **dir-access** ：（省略可）指定した名前のユーザを対象に、CSS ディレクトリへのアクセス権を指定する。CSS の 7 つのディレクトリ、つまり script、log、root（インストール済み CSS ソフトウェア）、archive、release root（設定ファイル）、core、MIB の各ディレクトリには、この順序でアクセス権が割り当てられています。デフォルトでは、7 つのディレクトリすべてに対して読み取りと書き込みの両方の権限（B）がユーザに与えられます。管理者や技術者は **dir-access** オプションによって、これらの各ディレクトリへの一連のアクセス権を、ユーザごとに設定できます。アクセス権の変更は、ディレクトリ関連の CLI コマンドの使用にも影響を与えます。

  dir-access オプションを使用するには、その前に **restrict user-database** コマンドを実行し、CSS ユーザ データベースにセキュリティ制限を設定する必要があります。

- *access* ：指定した名前のユーザを対象に、ディレクトリへのアクセス権を指定する。デフォルトでは、7 つのディレクトリすべてに対して読み取りと書き込みの両方の権限（B）がユーザに与えられます。これらの各ディレクトリへのアクセス権を表す次のコードを、連結した文字列として入力します。
  - **R**：CSS ディレクトリへの読み取り専用アクセス
  - **W**：CSS ディレクトリへの書き込み専用アクセス
  - **B**：CSS ディレクトリへの読み取りおよび書き込みを許可
  - **N**：CSS ディレクトリへのアクセスを許可しない

図 1-1 に、ユーザのディレクトリへのアクセス権の例を示します。

図 1-1　CSS ディレクトリへのアクセス権



たとえば、ユーザ名 *picard* のディレクトリへのアクセス権を設定するには、次のように入力します。

```
(config)# username picard password "captain" superuser NWBNNNR
```

既存のユーザ名のリストを表示するには、次のように入力します。

```
(config)# username ?
```

既存のユーザ名を削除するには、次のように入力します。

```
(config)# no username picard
```

ユーザのパスワードを変更するには、**username** コマンドを実行して、新しいパスワードを指定します。ユーザのステータスが SuperUser の場合には、superuser オプションも忘れずに指定してください。たとえば、次のように設定します。

```
(config)# username picard password "flute" superuser
```

注意　**no username** コマンドはユーザを完全に削除します。このコマンドは、いったん実行すると元に戻せないため、注意して使用してください。

CSS へのリモート アクセスの制御

CSS へのアクセスを制御するには、リモート（仮想）ユーザまたはコンソール ユーザを認証するように CSS を設定します。CSS では、ローカル ユーザ データベース、RADIUS サーバ、または TACACS+ サーバを使用してユーザを認証できます。また、認証を行わずにユーザ アクセスを許可したり、すべてのリモート ユーザを許可しないようにしたりすることもできます。

認証方式は、最大 3 種類（プライマリ、セカンダリ、またはターシャリ）まで設定できます。プライマリ認証方式が最初に使用されます。プライマリ認証方式が失敗すると（たとえば、RADIUS サーバがダウンしているか到達不能）、セカンダリ方式が使用されます。セカンダリ認証方式が失敗した場合は、ターシャリ認証方式が使用されます。ターシャリ認証方式も失敗した場合は、認証エラー メッセージが表示されます。

次の条件下ではセカンダリ方式もターシャリ認証方式も使用されません。

- 認証方式が **local** であり、ローカル ユーザ名がローカル ユーザ データベースにない。
- 認証方式が **local** であり、ローカル ユーザ名がローカル ユーザ データベースにあるが、パスワードが無効である。
- 認証方式が **radius** であり、RADIUS サーバが CSS からのプライマリ認証要求を拒否する。
- 認証方式が **tacacs** であり、TACACS+ サーバが CSS のプライマリ認証要求を拒否する。

RADIUS または TACACS+ を仮想認証方式またはコンソール認証方式で使用するには、先に RADIUS または TACACS+ セキュリティ サーバとの通信を可能にしておく必要があります。これには **radius-server** コマンド（第 3 章 RADIUS サーバのクライアントとしての CSS の設定参照）または **tacacs-server** コマンド（第 4 章 TACACS+ サーバのクライアントとしての CSS の設定参照）を使用します。

ここでは、次の内容について説明します。

- 仮想認証の設定
- コンソール認証の設定

仮想認証設定およびコンソール認証設定を表示するには、**show user-database** コマンドを使用します。

仮想認証の設定

仮想認証では、FTP、Telnet、SSHD、または CiscoView Device Manager（CVDM）インターフェイスを使用しているリモート ユーザがユーザ名とパスワードを使用して CSS にログインできます。また、ユーザ名とパスワードを使用しなくてもログインすることができます。CSS ですべてのリモート ユーザのアクセスを拒否することもできます。

CSS では、ローカル ユーザ データベース、RADIUS サーバ、または TACACS+ サーバを使用してユーザを認証するように設定できます。デフォルトでは、ローカル ユーザ データベースがユーザのプライマリ認証方式として使用されます。セカンダリ認証方式とターシャリ認証方式ではユーザ アクセスを禁止します。

プライマリ、セカンダリ、ターシャリのいずれかの仮想認証方式を設定するには、**virtual authentication** コマンドを使用します。このグローバル設定コマンドのシンタックスは次のとおりです。

virtual authentication [**primary**|**secondary**|**tertiary**
[**local**|**radius**|**tacacs**|**disallowed**]]

このコマンドのオプションは次のとおりです。

- **primary** ：CSS で最初に使用する認証方式を定義する。デフォルトのプライマリ仮想認証方式は、ローカル ユーザ データベースです。
- **secondary** ：CSS で最初の認証方式が失敗した場合に次に使用する認証方式を定義する。デフォルトのセカンダリ仮想認証方式では、すべてのユーザ アクセスを禁止します。

（注） TACACS+ サーバをプライマリ認証方式として設定する場合は、セカンダリ認証方式（**local** など）を定義する必要があります。

- **tertiary** ：CSS で 2 番目の認証方式が失敗した場合に次（3 番目）に使用する認証方式を定義する。デフォルトのターシャリ仮想認証方式では、すべてのユーザ アクセスを禁止します。
- **local** ：認証にローカル ユーザ データベースを使用する。
- **radius** ：認証に設定済みの RADIUS サーバを使用する。
- **tacacs** ：認証に設定済みの TACACS+ サーバを使用する。

- **disallowed** ：すべてのリモート ユーザのアクセスを禁止する。このオプションを指定しても、既存の接続は終了しません。

すでに CSS にログインしているユーザを削除するには、**disconnect** コマンドを使用します。

TACACS+ サーバをプライマリ仮想認証方式として定義するには、次のように入力します。

```
#(config) virtual authentication primary tacacs
```

ローカル ユーザ データベースをセカンダリ仮想認証方式として定義するには、次のように入力します。

```
#(config) virtual authentication secondary local
```

コンソール認証の設定

コンソール認証では、ユーザがユーザ名とパスワードを使用して、コンソール ポートに接続された端末経由で CSS にログインできるように設定できます。また、ユーザのユーザ名とパスワードがなくてもログインできるように設定することも可能です。CSS では、プライマリ認証方式でユーザ アクセスを禁止することができません。ただし、セカンダリ認証方式またはターシャリ認証方式では、ユーザ アクセスを禁止できます。

CSS では、ローカル ユーザ データベース、RADIUS サーバ、または TACACS+ サーバを使用してユーザを認証するように設定できます。デフォルトでは、ローカル ユーザ データベースがユーザのプライマリ認証方式として使用されます。セカンダリ認証方式とターシャリ認証方式ではユーザ アクセスを禁止します。

プライマリ、セカンダリ、ターシャリのいずれかのコンソール認証方式を設定するには、**console authentication** コマンドを使用します。このグローバル設定コマンドのは次のとおりです。

console authentication [**primary** [**local**|**radius**|**tacacs**|**none**]
|**secondary**|**tertiary** [**local**|**radius**|**tacacs**|**none**|**disallowed**]]

このコマンドのオプションは次のとおりです。

- **primary** ：CSS で最初に使用する認証方式を定義する。デフォルトのプライマリ コンソール認証方式は、ローカル ユーザ データベースです。

- **local** ：認証にローカル ユーザ データベースを使用する。
- **radius** ：認証に設定済みの RADIUS サーバを使用する。
- **tacacs** ：認証に設定済みの TACACS+ サーバを使用する。
- **none** ：認証方式を使用しない。すべてのユーザが CSS にアクセスできます。
- **secondary** ：CSS で最初の認証方式が失敗した場合に次に使用する認証方式を定義する。デフォルトのセカンダリ コンソール認証方式では、すべてのユーザ アクセスを禁止します。

（注） TACACS+ サーバをプライマリ認証方式として設定する場合は、セカンダリ認証方式（**local** など）を定義する必要があります。セカンダリ認証方式を設定せずに、デフォルトの **disallowed** を使用すると、CSS にログインできない可能性があります。

- **tertiary** ：CSS で 2 番目の認証方式が失敗した場合に次（3 番目）に使用する認証方式を定義する。デフォルトのターシャリ コンソール認証方式では、すべてのユーザ アクセスを禁止します。
- **disallowed** ：すべてのユーザのアクセスを禁止する（セカンダリまたはターシャリ認証方式のみ）。このオプションを指定しても、既存の接続は終了しません。

すでに CSS にログインしているユーザを削除するには、**disconnect** コマンドを使用します。

TACACS+ サーバをプライマリ コンソール認証方式として定義するには、次のように入力します。

```
#(config) console authentication primary tacacs
```

ローカル ユーザ データベースをセカンダリ コンソール認証方式として定義するには、次のように入力します。

```
#(config) console authentication secondary local
```

コンソール ポートで認証を無効にして、ユーザがユーザ名とパスワードを指定せずに CSS にアクセスできるようにするには、次のように入力します。

```
#(config) no console authentication
```

CSS への管理アクセスの制御

デフォルトでは、コンソール、FTP、SSH、SNMP および Telnet を使ったアクセスが有効に設定されています。CSS では、最大でそれぞれ 4 つの FTP セッションと Telent セッションがサポートされています。コンソール、FTP、SNMP、SSH、Telnet、ユーザ データベース、保護された XML と保護されていない XML、および CVDM による CSS へのデータ転送を有効または無効にするには、**restrict** および **no restrict** コマンドを使用します。

restrict コマンドを指定しても、CSS はアクセス制限されたポートでの接続試行を傍受します。TCP 接続の場合、CSS は TCP 3 ウェイ ハンドシェイクが完了した後でエラーで接続を終了し、データが転送されないようにします。UDP SNMP 接続の場合は、単にパケットを廃棄します。

制限付きポートを不正アクセスから保護するには、通常のトラフィックは CSS 内を通過させ、これらのポート宛てのパケットは拒否するように access control list （ACL; アクセス コントロール リスト）句を設定します。また、ACL を使用して CSS 自体を保護することもできます。CSS への ACL の設定方法については、「アクセス コントロール リストによる CSS ネットワーク トラフィックの制御」を参照してください。

CSS への管理アクセスの有効化

CSS へのコンソール、FTP、SNMP、SSH、Telnet、ユーザ データベース、保護された XML と保護されていない XML、CVDM アクセスを有効にするには、次の各 **no restrict** コマンドを使用します。

- **no restrict console** ：CSS へのコンソール アクセスを有効にする。デフォルトでは、有効に設定されています。
- **no restrict ftp** ：CSS への FTP アクセスを有効にする。デフォルトでは、有効に設定されています。
- **no restrict ssh** ：CSS への SSH アクセスを有効にする。デフォルトでは、有効に設定されています。
- **no restrict snmp** ：CSS への SNMP アクセスを有効にする。デフォルトでは、有効に設定されています。
- **no restrict telnet** ：CSS への Telnet アクセスを有効にする。デフォルトでは、有効に設定されています。

- **no restrict user-database** ：ユーザによる running-config ファイルの削除、およびユーザ名の作成や変更を有効にする。これらの操作は、管理者ユーザと技術者ユーザだけに許可されています。デフォルトでは、有効に設定されています。
- **no restrict secure-xml**：保護された HTTPS SSL 接続による CSS への XML 設定ファイルの転送を有効にする。デフォルトでは、無効に設定されています。
- **no restrict xml**：保護されていない HTTP 接続による CSS への XML 設定ファイルの転送を有効にする。デフォルトでは、無効に設定されています。
- **no restrict web-mgmt** ：CSS への CiscoView Device Manager（CVDM）からのアクセスを有効にする。デフォルトでは、無効に設定されています。

（注） Secure Shell Host（SSH; セキュア シェル ホスト）サーバを使用する場合は、Telnet アクセスを無効にします。SSH の設定については、第 2 章「SSHD プロトコルの設定」を参照してください。

たとえば、CVDM ユーザのアクセスを有効にするには、次のように入力します。

```
(config)# no restrict web-mgmt
```

Simple Network Management Protocol（SNMP; 簡易ネットワーク管理プロトコル）の設定についての詳細は、『*Cisco Content Services Switch Administration Guide*』を参照してください。XML を使用して、CSS に Web ベースでの設定変更を行う方法については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

CSS への管理アクセスの無効化

CSS へのコンソール、FTP、SNMP、SSH、Telnet、ユーザ データベース、保護された XML と保護されていない XML、CVDM アクセスを無効にするには、次の各 **restrict** コマンドを使用します。

- **restrict console** ：CSS へのコンソール アクセスを無効にする。デフォルトでは、有効に設定されています。
- **restrict ftp** ：CSS への FTP アクセスを無効にする。デフォルトでは、有効に設定されています。
- **restrict snmp** ：CSS への SNMP アクセスを無効にする。デフォルトでは、有効に設定されています。
- **restrict ssh** ：CSS への SSHD アクセスを無効にする。デフォルトでは、有効に設定されています。
- **restrict telnet** ：CSS への Telnet アクセスを無効にする。デフォルトでは、有効に設定されています。
- **restrict user-database** ：ユーザによる running-config ファイルの削除や、ユーザ名の作成、変更ができないようにする。これらの操作は、管理者ユーザと技術者ユーザだけに許可されています。デフォルトでは、有効に設定されています。
- **restrict secure-xml** ：保護された HTTPS SSL 接続による CSS への XML 設定ファイルの転送を無効にする。デフォルトでは、無効にされています。
- **restrict xml** :保護されていない HTTP 接続による CSS への XML 設定ファイルの転送を無効にする。デフォルトでは、無効にされています。
- **restrict web-mgmt** ：CSS への CVDM アクセスを無効にする。デフォルトでは、無効に設定されています。

たとえば、Telnet アクセスを無効にするには、次のように入力します。

```
(config)# restrict telnet
```

アクセス コントロール リストによる CSS ネットワーク トラフィックの制御

CSS には、アクセス コントロール リスト（ACL）を使用したトラフィックのフィルタリング機能が用意されています。ACL では、CSS のインターフェイスでパケットを転送するかブロックするかを制御することにより、着信ネットワーク トラフィックをフィルタ処理します。ACL は、ルーティング対象のネットワーク プロトコルに対して設定することができます。これにより、そのプロトコルのパケットが CSS を通過するときに、それらのパケットをフィルタリングすることができます。

ここでは、ACL の設定方法について説明します。

ACL の概要

CSS に ACL を設定すると、ネットワークへのアクセスに対して基本レベルのセキュリティが確立されます。ACL を設定していない CSS では、VLAN 回線を経由するすべてのパケットがネットワークに入ってくる可能性があります。ACL を使用すると、たとえば、CSS 回線に入ってくるすべての電子メール トラフィックを許可し、Telnet トラフィックをブロックするようなことが可能です。また、ACL を使用することにより、あるクライアントに対してネットワークの一部へのアクセスを許可し、別のクライアントに対して同じ領域へのアクセスを拒否することもできます。

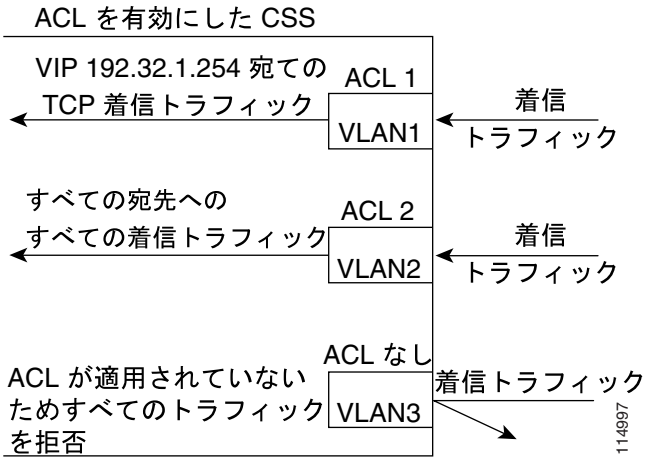
ACL は、ユーザ定義の句から構成されます。CSS では、これらの句を使用して、VLAN 回線での各パケットの処理方法を決定します。CSS は各パケットを検査し、パケットが ACL の句に一致するかどうかに基づいてそれを転送またはブロックします。トラフィックが回線を通過できるようにするには、ACL に permit 句を設定する必要があります。ACL の最後には、暗黙的な「deny all」句があります。

CSS に ACL を設定する際には、CSS の各 VLAN について、入ってくるトラフィックを制御するために ACL を適用する必要があります。回線に ACL を適用すると、ACL とその句が、その回線に割り当てられます。

ACL を各 CSS 回線に適用した後に、CSS で ACL を有効化する必要があります。ACL をグローバルに有効化すると、CSS のすべての回線に適用されます。ACL を有効化すると、すべての ACL に含まれる句が使用されて、すべての回線でトラフィックが許可または拒否されます。ACL が設定されていない回線には、暗黙的な「deny all」が適用され、この回線のすべてのトラフィックが拒否されます。

例として、図 1-2 に CSS の 3 つの VLAN 回線を示します。

図 1-2　CSS で有効化された ACL



VLAN1 で、宛先 VIP アドレス 192.32.1.254 へのすべての TCP トラフィックを許可するには、ACL 1 を作成し、次のように句を設定します。

clause 15 permit tcp any destination 192.32.1.254

その後に ACL 1 を VLAN1 に適用します。

VLAN2 で、任意の宛先へのすべてのトラフィックを許可するには、ACL 2 を作成し、次のように句を設定します。

clause 15 permit any any destination any

その後に ACL 2 を VLAN2 に適用します。

CSS で ACL を有効にすると、ACL に設定した permit 句の定義どおりに、VLAN1 と VLAN2 のトラフィックが許可されます。VLAN3 には ACL が適用されていないので、この回線には暗黙的な「deny all」が適用され、この回線のすべてのトラフィックが拒否されます。

注意　ACL は一種のファイアウォールとして機能し、セキュリティを確保します。*ACL を有効にする前に*、まず各 CSS 回線にトラフィックを許可する ACL を設定することが非常に重要です。トラフィックをまったく許可しない場合、そのネットワークの接続性が失われます。ただし接続が失われても、コンソール ポートには影響しません。

ACL をグローバルに有効化すると、各回線に ACL が割り当てられているかどうかに関係なく、*すべての* CSS 回線のすべてのトラフィックが影響を受けます。ACL を有効化すると、個々の ACL の permit 句で設定されていない回線のトラフィックはすべて*拒否されます*。各回線に ACL を適用していない場合、その回線へのトラフィックは拒否されます。

CSS で ACL を使用する場合、CSS のハードウェアには、レイヤ 3 またはレイヤ 4 の簡単な句を使用できる最大 10 個の ACL が実装されています。CSS ソフトウェアには、より複雑なレイヤ 5 の句を使用できる ACL が実装されています。

（注）　ACL は、CSS イーサネット管理ポートではサポートされません。

ACL は、ARP パケットをブロックしません。

ソース グループを指定した ACL 句を使用して、SSL モジュール宛てのトラフィックのソース アドレス変換を行うことはできません。CSS はこの句を受け入れますが、SSL モジュールで終了するフローがあってもこの句を無視します。SSL 処理後にサーバに向かう接続に対して NAT を適用することができます。

パッシブ FTP サーバのロード バランシングを実行しており、ACL を使用してソース グループを適用する場合は、そのソース グループに直接サービスを設定する必要があります。ソース グループによる FTP セッションのサポートの詳細は、『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。

ACL 設定のクイック スタート

表 1-1 に示すクイックスタートの手順を使用して、ACL を設定します。それぞれの手順に、作業を実行するために必要な CLI コマンドも示します。各機能の詳細については、この手順の後に示す各項を参照してください。

（注）　各 CSS 回線に設定する ACL には、1 つ以上の permit 句を含める必要があります。permit 句を 1 つも指定しないと、その回線へのすべてのトラフィックが拒否されます。

表 1-1　ACL 設定のクイック スタート

作業とコマンドの例
1. グローバル設定モードに入ります。 `# config` `(config)#`
2. ACL を作成して ACL モードにアクセスします。ACL インデックス番号を 1 ～ 99 の範囲で入力します。 `(config)# acl 7` `Create ACL <7>, [y,n]:y` `(config-acl[7])#`

表 1-1　ACL 設定のクイック スタート（続き）

作業とコマンドの例
3. ACL に句を設定します。CSS は、これらの句を、ACL を適用した回線（VLAN1 など）へのトラフィックを制御するために使用します。1 ～ 254 の句の番号を入力し、clause パラメータを定義します。句を定義するためのシンタックスは次のとおりです。 **clause** *number* **permit**|**deny**|**bypass** *protocol* [*source_info* {*source_port*}] **dest** [*dest_info* {*dest_port*}] {**log**} {**prefer** *servicename*} {**sourcegroup** *name*} **clause** コマンドのオプションについては、表 1-2 を参照してください。たとえばポート 20 ～ 23 で、ネットワークの外部から CSS の 1 つの回線へのすべてのユーザ アクセスをブロックするには、次のように入力します。 `(config-acl[7])# clause 10 deny any any destination range 20 23` CSS のその回線を通るその他のすべてのトラフィックを許可するには、次のように入力します。 `(config-acl[7])# clause 15 permit any any destination any`
4. ACL を個別の回線に適用します。この例では、VLAN は 1 つだけ（デフォルトの VLAN1）です。たとえば、acl7 を回線 VLAN1 に適用するには、次のように入力します。 `(config-acl[7])# apply circuit-(VLAN1)` **apply all** コマンドを使用して、ACL 7 を CSS のすべての回線に適用することもできます。
5. その他のすべての回線についてステップ 1 ～ 4 を繰り返し、1 つ以上の permit 句を含む ACL を作成し、これらの回線に適用します。CSS で ACL を有効にした場合、ACL が適用されていない回線へのトラフィックは拒否されます。

表 1-1　ACL 設定のクイック スタート（続き）

作業とコマンドの例
6. CSS のすべての ACL を有効にします。すべての ACL に対してグローバルな **acl enable** コマンドを入力すると、CSS のすべての回線に適用されます。 **注意** ACL をグローバルに有効化すると、CSS のすべての回線へのすべてのトラフィックが対象となるので、個々の ACL 内の permit 句に指定した回線のトラフィックだけが許可されます。ACL を適用していない回線には、暗黙的な「deny all」が適用され、この回線へのすべてのトラフィックが拒否されます。 次に例を示します。 `(config)# acl enable`

次の実行設定例は、表 1-1 で説明したコマンドを入力した結果を示しています。

```
!**************************** ACL ****************************
acl 7
    clause 10 deny any any destination range 20 23
    clause 15 permit any any destination any
  apply circuit-(VLAN1)

!************************* GLOBAL **************************
acl enable
```

ACL の作成

ACL には、CSS の回線上のトラフィックを制御する句を記述します。CSS で ACL をグローバルに有効化すると、すべての回線に適用されるので、各回線について ACL を作成する必要があります。1 つの ACL を、複数の回線に適用することが可能です。また、1 つの ACL を CSS のすべての回線に適用することもできます。

（注） ACL が設定されていない回線には、暗黙的な「deny all」が適用され、この回線のすべてのトラフィックが拒否されます。

ACL を作成して ACL モードにアクセスするには、**acl** *index number* コマンドを使用します。1 ～ 99 のインデックス番号で ACL を定義します。既存の ACL のリストを表示するには、**acl ?** コマンドを使用します。

```
(config)# acl 7
```

このモードにアクセスすると、プロンプトは作成したインデックス番号の ACL モードに変わります。たとえば、次のように入力します。

```
(config-acl[7])#
```

ACL を作成した後、句を追加する必要があります。詳細については、「句の設定」を参照してください。

ACL の削除

ACL とその句が不要になった場合は、その ACL を CSS から 削除できます。ACL を削除すると、ACL 内のすべての句も削除されます。ACL を削除するには、**no acl** コマンドを使用します。たとえば、ACL 7 を削除するには、次のように入力します。

```
(config)# no acl 7
```

CSS で ACL が有効化されている場合に、現在、回線に適用されている特定の ACL を削除すると、その ACL がその回線から削除され、CSS のその回線へのトラフィックは拒否されるようになります。この回線へのトラフィックを許可したい場合は、CSS で ACL をグローバルに無効化します。これにより、その回線へのすべてのトラフィックが許可されます。

たとえば、次のように操作します。

1. グローバル設定モードで、CSS のすべての ACL を無効化します。

   ```
   (config)# acl disable
   ```

2. ACL モードで、その回線から ACL を削除します。次のように入力します。

   ```
   (config-acl[7])# remove circuit-(VLAN1)
   ```

3. グローバル設定モードで、ACL を削除します。次のように入力します。

```
(config)# no acl 7
```

4. その回線に別の ACL を適用します。回線に ACL を適用していない場合に CSS でグローバルに ACL を有効にすると、CSS のその回線へのトラフィックが拒否されます。
5. CSS のすべての ACL を再度有効にします。次のように入力します。

```
(config)# acl enable
```

句の設定

ACL に設定した句によって、CSS で回線のトラフィックがどのように処理されるかが決定されます。句を設定する際には、句に番号を割り当てる必要があります。各句に割り当てる番号は重要です。CSS では、ACL を、句 1 から順に処理していきます。句に番号を割り当てる際は、最も詳細な一致条件の句に若い番号を割り当てます。一致条件が一般的になるにつれ、大きい値を割り当てます。

句は、番号順に入力する必要はありません。CSS では、句は適切な順序で ACL に自動挿入されます。たとえば、句 10 と句 24 を入力した後に句 15 を挿入すると、これらの句は正しい順序で挿入されます。

回線へのトラフィックを許可、拒否、またはバイパスする句を作成するには、**clause** コマンドを使用します。句*番号*は、句に割り当てる番号です。1 ～ 254 の番号を入力します。

（注） CSS で ACL が有効化されているときに ACL に新しい句を追加した場合は、その回線にその ACL を再度適用する必要があります。詳細については、「ACL をグローバルに有効化した場合の句の追加」の項を参照してください。

作成した句は、修正できません。句を修正するには、いったんその句を削除して、新しい句を作成する必要があります。句の削除の詳細は、「句の削除」の項を参照してください。

CSS は、すべての ACL に対し、255 番目の句として暗黙のデフォルトの「deny all」句を適用します。管理トラフィックを含むトラフィックは、permit 句を指定して許可するする必要があります。

clause コマンドのシンタックスは次のとおりです。

- **clause** *number* **bypass** ：回線へのトラフィックを*許可*し、そのトラフィックに適用されるコンテンツ ルールをバイパスする（処理しない）ための句を作成する。**clause bypass** のシンタックスは次のとおりです。

  clause *number* **bypass** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

（注）　**bypass** オプションでは、コンテンツ ルールへのトラフィック*だけを*バイパスするので、Network Address Translating（NATing; ネットワーク アドレス変換）は行われません。ソース グループを指定する ACL 句では、**bypass** オプションを使用しないでください。**bypass** オプションは、ソース グループの NATing には影響しません。

- **clause** *number* **deny** ：回線へのトラフィックを拒否するための句を ACL に作成する。**clause deny** のシンタックスは次のとおりです。

  clause *number* **deny** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

- **clause** *number* **permit** ：回線へのトラフィックを許可するための句を ACL に作成する。ACL に permit 句を設定すると、permit 句で指定されていないすべてのトラフィックは、デフォルトで拒否されます。**clause permit** のシンタックスは次のとおりです。

  clause *number* **permit** *protocol* [*source_info* {*source_port*}]
  dest [*dest_info* {*dest_port*}] {**sourcegroup** *name*} {**prefer** *servicename*}

（注）　ACL 句内の宛先がレイヤ 5 コンテンツ ルールの場合、CSS は接続をスプーフしないため、ACL 句は予想したとおりに機能しません。これを解決するために、TCP/IP アドレスとポートを許可する追加の句を設定することができます。この場合、両方の句でコンテンツが一致します。たとえば、次のようになります。
clause 14 permit any any destination content Layer5/L5 eq 80（元の句）
clause 15 permit tcp any destination 200.200.200.200 eq 80（これは、SYN を処理する追加の句です。この句では宛先の IP アドレスがレイヤ 5 コンテンツ ルールに設定されている IP アドレスになっています。この句の番号には、宛先のコンテンツの句の番号よりも大きい値を指定する必要があります。）

表 1-2 に、**clause** コマンドの変数とオプションを示します。ボールド体のシンタックスは、コマンド行に入力するキーワードを表します。イタリック体は、値を入力する変数（IP アドレスやホスト名など）を表します。

表 1-2　clause コマンドのオプション

変数とオプション	パラメータ
number	句に割り当てる番号。1 ～ 254 の番号を入力します。
action	句に割り当てるアクション。**bypass**、**deny**、**permit** のいずれかを入力します。
protocol	トラフィックのタイプに対応するプロトコル。**any**、**icmp**、**igp**、**igmp**、**ospf**、**tcp**、**udp** のいずれかを入力します。
source_info	トラフィックの発信元。次のいずれかを入力します。 • *ip_address* ：発信元 IP アドレスとオプションのマスクの IP アドレス。*サブネット マスク*は、IP アドレス形式のみで指定可能（省略可）。 • *hostname* ：発信元のホスト名。ホスト名は、ニーモニック名形式で入力します。最初に CSS の DNS クライアントを設定して、CSS でのホスト名の変換を有効にします。 • **any** ：発信元 IP アドレスおよびホスト名情報の任意の組み合わせ。 • **nql** *nql_name* ：IP アドレスのリストで構成される既存の Network Qualifier List（NQL; ネットワーク修飾子リスト）

表 1-2　clause コマンドのオプション（続き）

変数とオプション	パラメータ
source_port	トラフィックのソース ポート。ソース ポートを指定しない場合、この句は、すべてのポート番号からのトラフィックを許可します。次のいずれかを入力します。 • **eq** *port* ：指定したポート番号と同じポート • **lt** *port* ：指定したポート番号より小さいポート • **gt** *port* ：指定したポート番号より大きいポート • **neq** *port* ：指定したポート番号と異なるポート • **range** *low high* ：ポート番号の範囲。1 ～ 65535 の範囲の番号を入力します。*low* と *high* の番号は、スペースで区切ります。
destination_info	トラフィックの宛先に関する情報。次のいずれかを入力します。 • **destination any** ：送信先に関する情報の任意の組み合わせ • **destination content** *owner_name/rule_name* ：所有者のコンテンツ ルール。所有者とルール名は、/ 文字で区切ります。 • **destination** *ip_address* ：送信先 IP アドレスとオプションのサブネット マスクの IP アドレス。*サブネット マスク*は、IP アドレス形式だけで入力します。CIDR アドレス形式は使用できません。 • **destination** *hostname* ：宛先のホスト名。*hostname* を使用するには、最初に CSS の DNS クライアントを設定して、CSS でのホスト名の変換を有効にします。 • **nql** *nql_name* ：ホストの IP アドレスで構成される既存の NQL。NQL の名前を入力します。

表 1-2　clause コマンドのオプション（続き）

変数とオプション	パラメータ
destination_port	宛先のポート。次のいずれかを入力します。ポート番号またはポート名（オプションを指定）を使用できます。 • **eq** *port* ：指定したポート番号と同じポート • **lt** *port* ：指定したポート番号より小さいポート • **gt** *port* ：指定したポート番号より大きいポート • **neq** *port* ：指定したポート番号と異なるポート • **range** *low high* ：ポート番号の範囲。1 ～ 65535 の範囲の番号を入力します。*low* と *high* の番号は、スペースで区切ります。 • *port names* ： 　－ **https** = ポート 443 Https 　－ **ldap** = ポート 389 Ldap 　－ **bgp** = ポート 179 Bgp 　－ **ntp** = ポート 123 Ntp 　－ **nntp** = ポート 119 Nntp 　－ **pop** = ポート 110 Pop 　－ **http** = ポート 80 Http 　－ **gopher** = ポート 70 Gopher 　－ **domain** = ポート 53 Domain 　－ **smtp** = ポート 25 Smtp 　－ **telnet** = ポート 23 Telnet 　－ **ftp** = ポート 21 Ftp 　－ **ftp-data** = ポート 20 Ftp-data 　－ **none** = なし 宛先ポートを指定しない場合、この句では、すべてのポートへのトラフィックが許可されます。

表 1-2　clause コマンドのオプション（続き）

変数とオプション	パラメータ
sourcegroup *name*	トラフィックの宛先ソース グループ。グループ名を入力します。ソース グループのリストを表示するには、次のように入力します。 `show group ?` （注）　**clause** *number* **bypass** コマンドは、ソース グループの NATing に影響しません。 ソース グループを指定した ACL 句を使用して、SSL モジュール宛てのトラフィックのソース アドレス変換を行うことはできません。CSS はこの句を受け入れますが、SSL モジュールで終了するフローがあってもこの句を無視します。SSL 処理後にサーバに向かう接続に対して NAT を適用することができます。

表 1-2　clause コマンドのオプション（続き）

変数とオプション	パラメータ
prefer *service_name*	トラフィックの宛先として、指定したサービスを他のサービスより優先させます。優先サービスを複数定義する場合は、各サービスをカンマ（,）で区切ります。サービスは、最大 2 つまで定義できます。 Application Peering Protocol（APP）セッションで習得したサービスを優先サービスに設定することはできません。APP で習得したリモート サービスは ap-redirect@192.168.138.118 の形式をとり、**show service summary** 画面に表示されます。ACL 句を設定すると、このサービスを優先サービスとして使用できません。起動設定にこの句を保存して CSS を再起動した場合は、起動障害が発生します。この時点では APP を通してこのサービスを習得していないためです。たとえば、次のように入力します。 `clause 10 permit any any destination any prefer ap-redirect@192.168.138.118` **（注）** 優先サービスが設定された ACL は、スティッキ性よりも優先されます。 1 つの句内にソース グループと優先サービスの両方を指定する場合、先にソース グループを指定してから優先サービスを指定する必要があります。

ACL に句を作成すると、その ACL を回線に適用できます。詳細については、「回線または DNS 問い合せへの ACL の適用」の項を参照してください。

ACL をグローバルに有効化した場合の句の追加

CSS で ACL がグローバルに有効化されているときに、既に適用されている ACL に新しい句を追加した場合、その句を有効にするには、**apply circuit** コマンドを使用し、その回線にその ACL を再度適用する必要があります。

たとえば、ACL 7 を VLAN1 に適用し、CSS で ACL をグローバルに有効化したとします。その後、ACL 7 に句を追加して、CSS でこの句を有効にするには、次のように入力します。

```
(config-acl[7])# clause 200 permit any any destination any
(config-acl[7])# apply circuit-(VLAN1)
```

句の削除

既存の句を変更するには、ACL からいったんその句を削除して、再度追加する必要があります。句を削除するには、**no clause** コマンドを使用します。たとえば、句 6 を削除するには、次のように入力します。

```
(config-acl[7]) no clause 6
```

回線に ACL が適用され、CSS で有効化されている場合、その CSS ではこれらの ACL を使用中であると見なします。使用中の ACL から句を削除することはできません。句を削除するには、適用されている ACL を回線から削除してから、句を削除し、その ACL を再度回線に適用します。

たとえば、回線 VLAN1 の ACL7 から句 6 を削除するには、次のように操作します。

1. ACL モードで、回線 VLAN1 から ACL 7 を削除します。次のように入力します。

   ```
   (config-acl[7]) remove circuit-(VLAN1)
   ```

2. 次のように入力して、句 6 を削除します。

   ```
   (config-acl[7]) no clause 6
   ```

3. 回線 VLAN1 に ACL 7 を再度適用します。次のように入力します。

   ```
   (config-acl[7]) apply circuit-(VLAN1)
   ```

（注）　適用されている ACL を回線から削除すると、この回線に暗黙的な「deny all」が適用され、この回線へのすべてのトラフィックが拒否されます。回線から適用されている ACL を削除し、CSS でその回線へのトラフィックを許可したい場合は、グローバル設定モードで **acl disable** コマンドを使用して、ACL をグローバルに無効化します。CSS のすべての ACL を無効化することにより、その CSS ではすべての回線へのすべてのトラフィックが許可されます。

回線または DNS 問い合せへの ACL の適用

ACL に句を設定したら、**apply** コマンドを使用してすべての回線、個別の回線、または DNS 問い合せに ACL を割り当てます。

（注）　適用されている ACL に新しい句を追加するには、**apply circuit** コマンドを使用して、その回線に ACL を再度適用します。これにより、追加した句が有効になります。

空の ACL を回線に適用することはできません。適用しようとすると、エラーメッセージ「`Cannot apply ACL for it has no clauses`」が表示されます。

この ACL モード コマンドのシンタックスとオプションは次のとおりです。

- **apply all** ：既存のすべての回線に ACL を適用する。たとえば、次のように入力します。

  ```
  (config-acl[7])# apply all
  ```

- **apply circuit - (***circuit_name***)** ：個別の回線に ACL を適用する。たとえば、acl7 を回線 VLAN1 に適用するには、次のコマンドを入力します。

  ```
  (config-acl[7])# apply circuit-(VLAN1)
  ```

  回線のリストを表示するには、**apply ?** を入力します。

- **apply dns** ：DNS 問い合せに ACL を追加する。

  ```
  (config-acl[7])# apply dns
  ```

add dns *domain_ name* コマンドを使用して CSS のコンテンツ ルールにドメイン名を設定すると、そのドメイン名の DNS 問い合せは、**apply dns** コマンドで設定された ACL と *一致します*。

ただし、CSS に **dns-server** コマンドを設定した場合に、**host** コマンドで CSS 上に設定されたドメイン名の DNS 問い合せを CSS が受信すると、この DNS 問い合せは、**apply dns** コマンドで設定された ACL と一致 *しません*。

ACL を適用し、CSS で無効にした後は、**acl enable** グローバル設定コマンドを入力して、CSS で ACL を有効にする必要があります。**acl enable** コマンドの詳細については、この章で後述する「CSS での ACL の有効化」を参照してください。

回線または DNS 問い合せからの ACL の削除

ACL から句を削除する場合、回線に適用した ACL を削除する場合、または DNS 問い合せから ACL を削除する場合は、回線からその ACL を削除します。すべての回線、特定の回線、または DNS 問い合せから ACL を削除するには、**remove** コマンドを使用します。この ACL モード コマンドのシンタックスとオプションは次のとおりです。

- **remove all** ：すべての回線から ACL を削除する。

```
(config-acl[7])# remove all
```

- **remove circuit - (***circuit_name***)** ：特定の回線から ACL を削除する。次に例を示します。

```
(config-acl[7])# remove circuit-(VLAN1)
```

削除可能な回線のリストを表示するには、**remove ?** を入力します。

- **remove dns** ：DNS 問い合せから ACL を削除する。次に例を示します。

```
(config-acl[7])# remove dns
```

回線から ACL を削除する前に、CSS で ACL をグローバルに無効化することをお勧めします。CSS で ACL が有効になっている場合にある回線から ACL を削除すると、その回線に暗黙的な「deny all」が適用され、この回線へのすべてのトラフィックが拒否されるようになります。この回線へのトラフィックが拒否されないようにするには、CSS ですべての ACL を無効にした後に、その回線から ACL を削除する必要があります。CSS ですべての ACL を無効化することにより、すべての回線へのすべてのトラフィックが許可されます。

たとえば、次のように操作します。

1. グローバル設定モードで、CSS のすべての ACL を無効化します。

   ```
   (config)# acl disable
   ```

2. ACL モードで、対象の回線から ACL を削除します。

   ```
   (config-acl[7])# remove circuit-(VLAN1)
   ```

3. ACL に変更を加えます。

   回線から ACL を削除した場合、その回線に permit 句を含む別の ACL を設定し、適用します。この操作を行わないと、CSS で ACL を再度有効にしたときに、その回線へのトラフィックが拒否されるようになります。

4. ACL を回線に再度適用します。

   ```
   (config-acl[7])# apply circuit-(VLAN1)
   ```

5. グローバル設定モードで、CSS のすべての ACL を再度有効化します。

   ```
   (config)# acl enable
   ```

CSS での ACL の有効化

ACL とその句を設定した後に、その ACL を各 CSS 回線に適用すると、すべての ACL をグローバルに有効化し、CSS で使用できるようになります。すべての ACL をグローバルに有効化すると、CSS のすべての回線へのすべてのトラフィックが影響を受け、permit 句が指定されている ACL が設定された回線へのトラフィックだけが許可されるようになります。

注意　*ACL を有効にする前に*、まず各 CSS 回線にトラフィックを許可する ACL を設定することが非常に重要です。ACL を有効化すると、すべての回線が対象になります。トラフィックをまったく許可しない場合は、ネットワークの接続性が失われます。ACL を有効化すると、個々の ACL の permit 句で設定されていない回線のトラフィックはすべて*拒否されます*。ACL が適用されていない回線には、暗黙的な「deny all」句が適用されます。

たとえば、CSS に 3 つの回線（VLAN1、VLAN2、および VLAN3）を設定したとします。次に、ACL を VLAN1 だけに設定したとします。ACL をグローバルに有効化すると、VLAN1 では、その ACL に基づいてトラフィックが流れますが、VLAN2 と VLAN3 には ACL が設定されていないため、これらの回線に暗黙的な「deny all」句が適用され、これらの回線へのパケットは廃棄されます。

CSS で ACL をグローバルに有効化する前に、コンソールにアクセスできることを確認してください。ACL の設定が原因でネットワーク接続が失われた場合でも、コンソール ポートには影響はありません。

acl enable グローバル設定コマンドを使用して、CSS ですべての ACL を有効化します。すべての ACL をグローバルに有効にするには、次のコマンドを入力します。

```
(config)# acl enable
```

CSS での ACL の無効化

ACL を追加、変更、または削除する場合、または ACL の句を削除する場合は、回線からその ACL を削除する前に、すべての ACL を CSS で無効にすることをお勧めします。ACL をグローバルに無効化する前に、ある回線から ACL を削除すると、その回線に暗黙的な「deny all」が適用され、この回線へのすべてのトラフィックが拒否されます。

（注） CSS で ACL をグローバルに無効化すると、CSS のすべての ACL が無効化され、すべての CSS 回線へのすべてのトラフィックが許可されるようになります。

CSS のすべての ACL をグローバルに無効にするには、次のコマンドを入力します。

```
(config)# acl disable
```

ACL の表示

show acl コマンドを使用して、アクセス コントロール リストおよびその句を表示します。**show acl** コマンドは、すべてのモードで使用できます。

回線に適用された ACL 句を表示すると、次の項目が示されます。

- **Content Hits** ：フローはクライアント / サーバ間の UDP および TCP パケットストリームとして定義できる。CSS が完全にフローを設定するには、クライアントおよびサーバから多数のパケットを受信する必要があります。フローが完全に設定される前に受信したこれらのパケットは、すべて ACL チェックを受ける必要があります。これにより、ACL コンテンツ ヒット カウンタが増加することがあります。
- **Router Hits** ：TCP と UDP 以外のパケットはすべて ACL チェックを受ける必要があるので、ACL ルータ ヒット カウンタが増加する。CSS で終端する UDP および TCP トラフィック（たとえば、Telnet または FTP セッション）すべてでも、ACL ルータ ヒットカウンタが増加します。
- **DNS Hits** : ACL の句が DNS 問い合せに適用された場合に、ACL 句に一致し通過した DNS フローのパケット数。DNS ルックアップの数をカウントする DNS ヒット カウンタが表示されます。

CSS が受信したそれぞれのパケットの ACL ヒットの合計数は、フロー タイプと ACL マッチの有無によって異なります。CSS は、ACL のフローが完全に設定されるまで、受信したすべてのパケットに対して ACL のチェックを実行します。いったん ACL フローが設定されると、CSS は受信したそのフローに関連する残りのパケットに対して ACL チェックを行いません。このため ACL ヒットのカウンタは増加しません。

このコマンドのシンタックスは次のとおりです。

- **show acl** ：すべての ACL とその句を表示する。
- **show acl** *index* ：指定した ACL インデックス番号の句を表示する（有効な番号は 1 ～ 99）。
- **show acl config** ：ACL のグローバルな設定を表示する。このコマンドでは、どの ACL がどの回線に適用されているかが示されます。

このコマンドは、次のように入力します。

```
(config)# show acl 2
```

表 1-3 に、**show acl** コマンドで表示されるフィールドを示します。

表 1-3　show acl コマンド出力のフィールド

フィールド	内容
Acl	ACL に割り当てられた番号（1 ～ 99）
Clause	句に割り当てられた番号（1 ～ 254）
Action	着信トラフィックを、句（permit、deny、または bypass）とそのトラフィック タイプのプロトコルで制御する方式
Source	設定されたトラフィックの発信元
Destination	設定されたトラフィックの送信先
Log	指定した句の ACL ロギングが有効または無効のどちらであるかを示します。
Content Hits	フローが設定されるまでに CSS が受信したパケットの増分カウント

表 1-3　show acl コマンド出力のフィールド（続き）

フィールド	内容
Router Hits	Telnet または FTP セッションで、あるいは TCP または UDP 以外のパケットから CSS に直接転送されたパケットの増分カウント
DNS Hits	DNS フローで ACL 句に一致するパケットの増分カウント

ACL カウンタ表示の 0 への設定

zero counts コマンドを使用して、特定の ACL に対して、**show acl** コマンド画面内のコンテンツと DNS ヒット カウンタをゼロにリセットします。このコマンドを実行するには、ACL のモードに入っている必要があります。このコマンドでは、その ACL のカウンタだけがクリアされます。

このコマンドのシンタックスとオプションは次のとおりです。

```
(config-acl[7])# zero counts
```

ACL アクティビティのロギング

ACL アクティビティをロギングするように設定すると、句と ACL に一致するパケットのイベントがロギングされます。ログ情報は、**logging** コマンドで指定した場所に送信されます。**logging** コマンドの詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

（注） ACL またはその句のロギングは、お勧めしていません。ACL またはその句のロギングを有効にすると、CSS のパフォーマンスが低下する可能性があります。特定の ACL 句にロギングを設定する前に、グローバルな ACL ロギングが有効になっていることを確認してください。グローバルな ACL ロギングを有効にするには、グローバル設定モードで **logging subsystem acl level debug-7** コマンドを使用します。
CSS は、**clause log enable** コマンドを実行設定に保存しないため、CSS を再度ブートした場合は、ロギングを再度有効にする必要があります。

既存の ACL 句のロギングを有効化するには、次のように **clause** コマンドに **log enable** オプションを指定します。

```
(config-acl[7])# clause 1 log enable
```

CSS で ACL をグローバルに有効化している場合、次のように既存の ACL 句のロギングを設定します。

1. グローバル設定モードで、CSS のすべての ACL を無効化します。

   ```
   (config)# acl disable
   ```

2. ロギングを有効にする対象の ACL モードに入ります。

   ```
   (config)# acl 7
   (config-acl[7])#
   ```

3. 回線から ACL を削除します。

   ```
   (config-acl[7]) remove circuit-(VLAN1)
   ```

4. 既存の句のロギングを有効にします。

   ```
   (config-acl[7])# clause 1 log enable
   ```

5. ACL を回線に再度適用します。

   ```
   (config-acl[7])# apply circuit-(VLAN1)
   ```

6. グローバル設定モードで、CSS のすべての ACL を再度有効化します。

   ```
   (config)# acl enable
   ```

特定の句の ACL ロギングを無効にするには、次のように入力します。

1. グローバル設定モードで、CSS のすべての ACL を無効化します。

   ```
   (config)# acl disable
   ```

2. ロギングを無効にする対象の ACL モードに入ります。

   ```
   (config)# acl 7
   (config-acl[7])#
   ```

3. 回線から ACL を削除します。

```
(config-acl[7]) remove circuit-(VLAN1)
```

4. 既存の句のロギングを無効にします。

```
(config-acl[7])# clause 1 log disable
```

5. ACL を回線に再度適用します。

```
(config-acl[7])# apply circuit-(VLAN1)
```

6. グローバル設定モードで、CSS のすべての ACL を再度有効化します。

```
(config)# acl enable
```

すべての ACL 句のロギングをグローバルに無効にするには、次のように入力します。

```
(config)# no logging subsystem acl
```

ACL の例

次の ACL では、1 本の VLAN （VLAN1）で CSS、Server1 および Server2 にセキュリティを設定します。この ACL は次のように動作します。

- サブネット 172.16.107.x からのクライアントに対し、さまざまなアプリケーション（Telnet、FTP、TFTP など）を使用して VLAN1 の Server1 および Server2 にアクセスすることを許可します。
- サブネット 172.16.107.x からのクライアントに対し、URL 172.16.107.35（VIP アドレス）でブラウザを起動することを許可します。
- 172.16.107.x　以外のサブネットにあるクライアントが、VLAN1、Server1、Server2 にアクセスできないようにします。

各句では、次のセキュリティが提供されます。

- 句 20 では、発信元のサブネット 172.16.107.0 から Server1 （IP アドレス 172.16.107.15）へのすべてのプロトコルを許可します。
- 句 30 では、発信元のサブネット 172.16.107.0 から Server2 （IP アドレス 172.16.107.16）へのすべてのプロトコルを許可します。

- 句 40 では、発信元のサブネット 172.16.107.0 から VIP アドレス 172.16.107.35 ポート 80（HTTP）へのすべてのプロトコルを許可します。
- 句 50 では、キープアライブを含む、Internet Control Message Protocol（ICMP; インターネット制御メッセージ プロトコル）のすべてのトラフィックに対し VLAN への双方向通信を許可します。キープアライブ サービスを使用している場合は、キープアライブ トラフィックを許可する句を設定する必要があります。
- 句 60 では、UDP を使用した VLAN のポート 520 への Routing Information Protocol（RIP; ルーティング情報プロトコル）のアップデートを許可します。この句は、使用中のルータが 172.16.107.x 以外のサブネットに存在する場合に必要です。
- 句 70 では、ACL で許可されていないすべてのトラフィックを拒否します。

```
!**************************** ACL ***************************
acl 1
clause 20 permit any 172.16.107.0 255.255.255.0 destination
172.16.107.15
clause 30 permit any 172.16.107.0 255.255.255.0 destination
172.16.107.16
clause 40 permit any 172.16.107.0 255.255.255.0 destination
172.16.107.35 eq 80
clause 50 permit ICMP any destination any
clause 60 permit udp any destination any eq 520
clause 70 deny any any destination any
apply circuit-(VLAN1)
```

ACL へのネットワーク修飾子リストの設定

NQL 設定モードでは、ネットワーク修飾子リスト（NQL）を設定することができます。NQL は、IP アドレスおよびサブネット マスクにより識別される、ネットワークまたは特定のサービスのリストです。NQL は ACL 句に発信元または送信先として割り当てます。複数のネットワークを NQL にグループ化して、その NQL を 1 つの ACL 句に割り当てると、そのグループにその 1 つの句を作成するだけで済みます。ネットワークごとに個別の句を作成する必要はありません。

CSS では、次のものを最大 512 個まで設定できます。

- NQL あたりのネットワークまたはサービス
- CSS あたりの NQL

この機能は、特定のネットワークをバイパスしてコンテンツ要求を元のサーバ（コンテンツが保存されているサーバ）に直接送信するキャッシング環境などで役立ちます。また、特定のネットワークに基づいて、あるサービスを優先させる場合にも NQL を使用できます。

NQL 設定モードにアクセスするには、**nql** コマンドを使用します。プロンプトは、(config-nql [name]) に変わります。NQL モードでこのコマンドを使用して他の NQL にアクセスすることもできます。

NQL の設定については、次の項を参照してください。

- NQL の作成
- NQL の説明の記述
- NQL へのネットワークの追加
- ACL 句への NQL の追加
- NQL 設定の表示

NQL の作成

作成する新しい NQL または既存の NQL の名前を入力します。名前は、スペースを含まない 31 文字以内のテキスト文字列を引用符で囲まずに入力します。CSS ごとに最大 512 個の NQL を作成できます。

たとえば、次のように入力します。

```
(config)# nql bypass_nql
(config-nql[bypass_nql])#
```

既存の NQL のリストを表示するには、**nql ?** を入力します。NQL が存在しない場合は、新しい名前を入力するよう指示されます。

既存の NQL を削除するには、**no nql** コマンドを使用します。たとえば、次のように入力します。

```
(config)# no nql bypass_nql
```

NQL の説明の記述

NQL の説明を記述するには、NQL モードで **description** コマンドを使用します。NQL の説明は、63 文字以内のテキスト文字列を引用符で囲んで入力します。

たとえば、次のように入力します。

```
(config-nql[bypass_nql])# description "Bypass services"
```

NQL へのネットワークの追加

最大 512 個のネットワークまたはサービスを NQL に追加するには、**ip address** コマンドを使用します。IP アドレスを、サブネット プレフィックスまたはサブネット アドレスと共に入力します。必要に応じて、IP アドレスの説明を追加したり、ロギングをオンにしたりすることもできます。

このコマンドのシンタックスおよびオプションは、次のとおりです。

ip address *ip_address*[*/subnet_prefix*| *subnet_mask*] {**"***description***"**} {**log**}

変数とオプションは、次のとおりです。

- *ip_address* ：送信先のネットワーク アドレス。IP アドレスをドット付き 10 進表記で入力します（例 192.168.0.0）。
- *subnet_prefix*|*subnet_mask* ：CIDR ビット数表記の IP サブネット マスク プレフィックス長（/16 など）。有効なプレフィックス長の範囲は 8 ～ 32 です。IP アドレスとプレフィックス長の間にはスペースを入れないでください。
- *subnet_mask* ：ドット付き 10 進表記の IP サブネット マスク（たとえば、255.255.0.0）
- **"***description***"** : IP アドレスの説明。63 文字以内のテキスト文字列を引用符で囲んで入力します。
- **log** ：NQL に関連するイベントのログ。このオプションを入力しない場合、イベントのログは記録されません。NQL イベントのログを記録するには、グローバルな NQL ロギングを有効にする必要があります。グローバルな NQL ロギングを有効にするには、**(config) logging subsystem nql level debug-7** コマンドを使用します。ロギングの詳細については、『*Cisco Content Services Switch Administration Guide*』を参照してください。

たとえば、NQL bypass_nql に 2 つのネットワークを追加するには、次のように入力します。

```
(config-nql[bypass_nql])# ip address 192.168.0.0/16 "Network of
dynamic mail content" log
(config-nql[bypass_nql])# ip address 123.123.123.0/24
```

ネットワークで発生したイベントのログを記録するには、グローバルな NQL ロギングを有効にする必要があります。たとえば、次のように入力します。

```
(config)# logging subsystem nql level debug-7
```

（注） エントリの作成時に説明を追加したり、ロギング機能を有効にしないで、これらの作業を後で行う場合は、最初にそのエントリを削除してから、希望のオプションを指定してこのエントリを再度追加してください。

NQL から IP アドレスを削除するには、**no ip address** コマンドを使用します。次に例を示します。

```
(config-nql[bypass_nql])# no ip address 192.168.0.0/16
```

ACL 句への NQL の追加

NQL を ACL 句に追加するには、次の手順に従います。

1. ACL を作成します。たとえば、次のように入力します。

```
(config)# acl 10
```

2. 発信元または宛先として NQL を含む句を定義します。

   次の句の例では、任意の発信元からポート 80 の NQL bypass_nql で定義された宛先のネットワークに発信されたトラフィックのコンテンツ ルールがバイパスされます。

```
(config-acl[10])# clause 1 bypass any any destination nql
bypass_nql eq 80
```

NQL 設定の表示

NQL 設定の情報を表示するには、**show nql** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

- **show nql** ：すべての NQL に関する情報を表示する。NQL モードでこのコマンドを入力すると、現在の NQL のアドレスだけが表示されます。
- **show nql** *nql_name* ：指定した NQL の情報を表示する。NQL 名は、大文字小文字を区別したスペースを含まないテキスト文字列を引用符で囲まずに入力します。既存の NQL 名のリストを表示するには、**show nql ?** コマンドを使用します。

たとえば、次のように入力します。

```
(config-nql[bypass_nql])# show nql
```

表 1-4 に、**show nql** コマンドで表示されるフィールドを示します。

表 1-4　show nql コマンド出力のフィールド

フィールド	内容
Name	NQL の名前
Description	NQL に関連付けられている説明
IP Addresses	NQL でサポートされる IP アドレスとサブネット マスク。説明が設定されている場合、アドレスの後に説明が表示されます。