ルート マップ

この章では、Cisco ASA にルート マップを設定およびカスタマイズする方法について説明します。

ルート マップについて

ルート マップは、ルートを OSPF、RIP、EIGRP、または BGP ルーティング プロセスに再配布するときに使用します。また、デフォルト ルートを OSPF ルーティング プロセスに生成するときにも使用します。ルート マップは、指定されたルーティング プロトコルのどのルートを対象ルーティング プロセスに再配布できるのかを定義します。

ルート マップは、広く知られた ACL と共通の機能を数多く持っています。両方に共通する主な特性は次のとおりです。

- いずれも、それぞれが許可または拒否の結果を持つ個々の文を一定の順序で並べたものです。ACL またはルートマップの評価は、事前に定義された順序でのリストのスキャンと、一致する各文の基準の評価で構成されています。リストのスキャンは、文の一致が初めて見つかり、その文に関連付けられたアクションが実行されると中断します。
- これらは一般的なメカニズムです。基準一致と一致解釈は、適用方法とこれらを使用する機能によって決定します。異なる機能に適用される同じルートマップの解釈が異なることがあります。

次のように、ルート マップと ACL には違いがいくつかあります。

- ルート マップは ACL よりも柔軟性が高く、ACL が確認できない基準に基づいてルートを確認できます。たとえば、ルートマップはルートのタイプが内部であるかどうかを確認できます。

- 設計規則により、各 ACL は暗黙の deny 文で終了します。一致試行の間にルート マップの終わりに達した場合は、そのルートマップの特定のアプリケーションによって結果が異なります。再配布に適用されるルートマップの動作はACLと同じです。ルートがルートマップのどの句とも一致しない場合は、ルートマップの最後に deny 文が含まれている場合と同様に、ルートの再配布が拒否されます。

permit 句と deny 句

ルート マップでは permit 句と deny 句を使用できます。deny 句は、ルートの照合の再配布を拒否します。ルートマップでは、一致基準として ACL を使用できます。ACL には permit 句と deny 句もあるので、パケットが ACL と一致した場合に次のルールが適用されます。

- ACL permit + route map permit：ルートは再配布されます。
- ACL permit + route map deny：ルートは再配布されません。
- ACL deny + route map permit or deny：ルート マップの句は一致せず、次のルート マップ句が評価されます。

match 句と set 句の値

各ルート マップ句には、次の 2 種類の値があります。

- match 値は、この句が適用されるルートを選択します。
- set 値は、ターゲット プロトコルに再配布される情報を変更します。

再配布される各ルートについて、ルータは最初にルート マップの句の一致基準を評価します。一致基準が満たされると、そのルートは、permit 句または deny 句に従って再配布または拒否され、そのルートの一部の属性が、set コマンドによって設定された値で変更されます。一致基準が満たされないと、この句はルートに適用されず、ソフトウェアはルートマップの次の句でルートを評価します。ルートマップのスキャンは、ルートと一致する句が見つかるまで、もしくはルートマップの最後に到達するまで続行します。

次のいずれかの条件が満たされる場合は、各句の match 値または set 値を省略したり、何回か繰り返したりできます。

- 複数の match エントリが句に含まれる場合に、特定のルートが句に一致するためには、そのルートですべての照合に成功しなければなりません（つまり、複数の match コマンドでは論理 AND アルゴリズムが適用される）。
- match エントリが 1 つのエントリの複数のオブジェクトを指している場合は、そのいずれかが一致していなければなりません（論理 OR アルゴリズムが適用される）。
- match エントリがない場合は、すべてのルートが句に一致します。
- ルート マップの permit 句に set エントリが存在しない場合、ルートは、その現在の属性を変更されずに再配布されます。

（注） ルート マップの deny 句では set エントリを設定しないでください。deny 句を指定するとルートの再配布が禁止され、情報が何も変更されないからです。

match エントリまたは set エントリがないルート マップ句はアクションを実行します。空の permit 句を使用すると、変更を加えずに残りのルートの再配布が可能になります。空の deny 句では、他のルートの再配布はできません。これは、ルートマップがすべてスキャンされたときに、明示的な一致が見つからなかったときのデフォルト アクションです。

ルート マップのガイドライン

ファイアウォール モード

ルーテッド ファイアウォール モードでだけサポートされています。トランスペアレント ファイアウォール モードはサポートされません。

その他のガイドライン

ルート マップは、ユーザ、ユーザ グループ、または完全修飾ドメイン名のオブジェクトを含む ACL をサポートしていません。

ルート マップの定義

ルート マップを定義する必要があるのは、指定したルーティング プロトコルからのどのルートを対象ルーティング プロセスに再配布できるのかを指定するときです。

手順

ルート マップのエントリを作成します。

route-map *name* {**permit** | **deny**} [*sequence_number*]

例：

```
ciscoasa(config)# route-map name {permit} [12]
```

ルート マップのエントリは順番に読み取られます。この順序は、*sequence_number* 引数を使用して指定できます。この引数で指定しなければ、ルートマップエントリを追加した順序がASA で使用されます。

ルート マップのカスタマイズ

ここでは、ルート マップをカスタマイズする方法について説明します。

特定の宛先アドレスに一致するルートの定義

手順

ステップ 1 ルート マップのエントリを作成します。

route-map *name* {**permit** | **deny**} [*sequence_number*]

例：

```
ciscoasa(config)# route-map name {permit} [12]
```

ルート マップのエントリは順番に読み取られます。この順序は、*sequence_number* オプションを使用して指定できます。この引数で指定しなければ、ルート マップ エントリを追加した順序が ASA で使用されます。

ステップ 2 標準 ACL またはプレフィックス リストに一致する宛先ネットワークを持つ任意のルートを照合します。

match ip address *acl_id* [*acl_id*] [...][prefix-list]

例：

```
ciscoasa(config-route-map)# match ip address acl1
```

複数の ACL を指定する場合、ルートは任意の ACL を照合できます。

ステップ 3 指定したメトリックを持つ任意のルートを照合します。

match metric *metric_value*

例：

```
ciscoasa(config-route-map)# match metric 200
```

metric_value には、0 ～ 4294967295 の範囲が指定できます。

ステップ 4 標準 ACL と一致するネクスト ホップ ルータ アドレスを持つ任意のルートを照合します。

match ip next-hop *acl_id* [*acl_id*] [...]

例：

```
ciscoasa(config-route-map)# match ip next-hop acl2
```

複数の ACL を指定する場合、ルートは任意の ACL を照合できます。

ステップ 5 指定されたネクスト ホップ インターフェイスを持つ任意のルートを照合します。

match interface *if_name*

例：

```
ciscoasa(config-route-map)# match interface if_name
```

2 つ以上のインターフェイスを指定する場合、ルートはいずれかのインターフェイスと一致します。

ステップ 6 標準の ACL と一致するルータによってアドバタイズされた任意のルートを照合します。

match ip route-source *acl_id* [*acl_id*] [...]

例：

```
ciscoasa(config-route-map)# match ip route-source acl_id [acl_id] [...]
```

複数の ACL を指定する場合、ルートは任意の ACL を照合できます。

ステップ 7 ルート タイプを照合します。

match route-type {**internal** | **external** [**type-1** | **type-2**]}

ルート アクションのメトリック値の設定

ルートが **match** コマンドで一致する場合は、次の **set** コマンドによって、ルートを再配布する前にルートで実行するアクションが決まります。

ルート アクションのメトリック値を設定するには、次の手順を実行します。

手順

ステップ 1 ルート マップのエントリを作成します。

route-map *name* {**permit** | **deny**} [*sequence_number*]

例：

```
ciscoasa(config)# route-map name {permit} [12]
```

ルート マップのエントリは順番に読み取られます。この順序は、*sequence_number* 引数を使用して指定できます。この引数で指定しなければ、ルートマップエントリを追加した順序が ASA で使用されます。

ステップ 2 ルート マップのメトリック値を設定します。

set metric *metric_value*

例：

```
ciscoasa(config-route-map)# set metric 200
```

metric_value の引数は、0～294967295 の範囲で指定できます。

ステップ 3 ルート マップのメトリック タイプを設定します。

set metric-type {**type-1** | **type-2**}

例：

```
ciscoasa(config-route-map)# set metric-type type-2
```

metric-type 引数には type-1 と type-2 があります。

ルート マップの例

次の例は、ホップ カウント 1 でルートを OSPF に再配布する方法を示しています。

ASA は、これらのルートをメトリック 5、メトリック タイプ 1 で外部 LSA として再配布します。

```
ciscoasa(config)# route-map 1-to-2 permit
ciscoasa(config-route-map)# match metric 1
ciscoasa(config-route-map)# set metric 5
ciscoasa(config-route-map)# set metric-type type-1
```

次に、メトリック値が設定された EIGRP プロセス 1 に 10.1.1.0 のスタティック ルートを再配布する例を示します。

```
ciscoasa(config)# route outside 10.1.1.0 255.255.255.0 192.168.1.1
ciscoasa(config-route-map)# access-list mymap2 line 1 permit 10.1.1.0 255.255.255.0
ciscoasa(config-route-map)# route-map mymap2 permit 10
ciscoasa(config-route-map)# match ip address mymap2
ciscoasa(config-route-map)# router eigrp 1
ciscoasa(config-router)# redistribute static metric 250 250 1 1 1 route-map mymap2
```

ルート マップの履歴

表 *1:* ルート マップの機能履歴

機能名	プラットフォーム リリース	機能情報
ルート マップ	7.0(1)	この機能が導入されました。 **route-map** コマンドが導入されました。
スタティックおよびダイナミック ルート マップのサポートの強化	8.0(2)	ダイナミックおよびスタティック ルート マップのサポートが強化されました。
ダイナミック ルーティング プロトコル（EIGRP、OSPF、RIP）のステートフル フェールオーバーと一般的なルーティング関連動作のデバッグのサポート	8.4(1)	**debug route**、および **show debug route** コマンドが導入されました。 **show route** コマンドが変更されました。
マルチ コンテキスト モードのダイナミック ルーティング	9.0(1)	ルート マップは、マルチ コンテキスト モードでサポートされます。
BGP のサポート	9.2(1)	この機能が導入されました。 **router bgp** コマンドが導入されました。
プレフィックス ルールの IPv6 サポート	9.3.2	この機能が導入されました。