CHAPTER 8

オブジェクトの設定

オブジェクトとは、コンフィギュレーションで使用するための再利用可能なコンポーネントです。オブジェクトは、インライン IP アドレスの代わりに ASA 1000V コンフィギュレーションで定義し、使用できます。オブジェクトを使用すると、コンフィギュレーションのメンテナンスが容易になります。これは、一箇所でオブジェクトを変更し、このオブジェクトを参照している他のすべての場所に反映できるからです。オブジェクトを使用しなければ、1 回だけ変更するのではなく、必要に応じて各機能のパラメータを変更する必要があります。たとえば、ネットワーク オブジェクトによって IP アドレスおよびサブネット マスクが定義されており、このアドレスを変更する場合、この IP アドレスを参照する各機能ではなく、オブジェクト定義でアドレスを変更することだけが必要です。

この章では、オブジェクトを設定する方法について説明します。説明する項目は次のとおりです。

オブジェクトとグループの設定

この項は、次の内容で構成されています。

オブジェクトとグループに関する情報

ASA 1000V では、オブジェクトおよびオブジェクト グループがサポートされています。必要に応じて、1 つまたは複数のオブジェクト グループのオブジェクトを関連付けるか、関連付けを解除して、オブジェクトが重複しないようにしつつ、必要に応じてオブジェクトを再利用できるようにします。

この項は、次の内容で構成されています。

オブジェクトに関する情報

オブジェクトは、任意のコンフィギュレーションでインライン IP アドレスの代わりに ASA 1000V によって作成され、使用されます。オブジェクトは、特定の IP アドレスおよびネットマスクのペアまたはプロトコル（およびオプションでポート）で定義できます。このオブジェクトは、複数のコンフィギュレーションで使用できます。利点は、この IP アドレスまたはプロトコルに対して作成されたコンフィギュレーションを変更する場合、実行コンフィギュレーションですべてのルールを変更する必要のないことです。オブジェクトを変更すると、指定したオブジェクトを使用するすべてのルールにこの変更が自動的に適用されます。オブジェクトは、ネットワーク オブジェクトとサービス オブジェクトの 2 種類を設定できます。これらのオブジェクトは、ネットワーク アドレス変換（NAT）、アクセス リスト、およびオブジェクト グループで使用できます。

オブジェクト グループに関する情報

類似オブジェクトをグループにまとめると、オブジェクトごとに ACE を入力する代わりに、ACE でオブジェクト グループを使用できるようになります。次のタイプのオブジェクト グループを作成できます。

- プロトコル
- ネットワーク
- サービス
- ICMP タイプ

たとえば、次の 3 つのオブジェクト グループを考えてみます。

- MyServices：内部ネットワークへのアクセスが許可されるサービス要求の TCP/UDP ポート番号を含む。
- TrustedHosts：最大範囲のサービスとサーバへのアクセスが許可されるホスト アドレスとネットワーク アドレスを含む。
- PublicServers：最大のアクセスが提供されるサーバのホスト アドレスを含む。

上記のグループを作成すると、1 つの ACE を使用して、信頼できるホストが公開サーバのグループにサービス要求を許可することが可能になります。

オブジェクト グループを他のオブジェクト グループにネストすることもできます。

オブジェクトとグループのガイドラインと制限事項

オブジェクト グループには、次のガイドラインと制限事項が適用されます。

- オブジェクトおよびオブジェクト グループは、同じ名前スペースを共有します。
- オブジェクト グループには、固有の名前が必要となります。「Engineering」という名前のネットワーク オブジェクト グループと「Engineering」という名前のサービス オブジェクト グループを作成する場合、少なくとも 1 つのオブジェクト グループ名の最後に識別子（または「タグ」）を追加して、その名前を固有のものにする必要があります。たとえば、「Engineering_admins」と「Engnineering_hosts」という名前を使用すると、オブジェクト グループの名前を固有のものにして特定可能にすることができます。
- 別のコマンドで使用されている場合は、オブジェクト グループを削除したり、オブジェクト グループを空にすることはできません。

- オブジェクト グループの一意の修飾名は、ASA 1000V の VNMC モードのみでサポートされます。

オブジェクトの設定

この項は、次の内容で構成されています。

ネットワーク オブジェクトの設定

ネットワーク オブジェクトには、1 つの IP アドレスとマスクのペアが含まれます。ネットワーク オブジェクトは、ホスト、サブネット、または範囲の 3 種類です。

オブジェクト定義の一部として、自動 NAT を設定することもできます。詳細については、第 12 章「ネットワーク オブジェクト NAT の設定」を参照してください。

手順の詳細

	コマンド	目的
ステップ 1	**object network** *obj_name* 例： hostname(config)# object-network OBJECT1	新しいネットワーク オブジェクトを作成します。*obj_name* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） プロンプトが、ネットワーク オブジェクト コンフィギュレーション モードに変わります。
ステップ 2	{**host** *ip_addr* | **subnet** *net_addr net_mask* | **range** *ip_addr_1 ip_addr_2*} 例： hostname(config-network-object)# host 10.2.2.2	ネームド オブジェクトに IP アドレスを割り当てます。ホスト アドレス、サブネット、またはアドレスの範囲を設定できます。
ステップ 3	**description** *text* 例： hostname(config-network-object)# description Engineering Network	オブジェクトに説明を追加します。

例

ネットワーク オブジェクトを作成するには、次のコマンドを入力します。

```
hostname (config)# object network OBJECT1
hostname (config-network-object)# host 10.2.2.2
```

サービス オブジェクトの設定

サービス オブジェクトには、プロトコル、およびオプションの送信元ポートまたは宛先ポート、あるいはその両方が含まれます。

手順の詳細

	コマンド	目的
ステップ 1	**object service** *obj_name* **例：** `hostname(config)# object-service SERVOBJECT1`	新しいサービス オブジェクトを作成します。*obj_name* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） プロンプトが、サービス オブジェクト コンフィギュレーション モードに変わります。
ステップ 2	**service** {*protocol* | **icmp** *icmp-type* | **icmp6** *icmp6-type* | {**tcp** | **udp**} [**source** *operator port*] [**destination** *operator port*]} 例： `hostname(config-service-object)# service tcp source eq www destination eq ssh`	送信元のマッピング アドレスのサービス オブジェクトを作成します。 *protocol* 引数には、IP プロトコルの名前または番号を指定します。 **icmp**、**tcp**、または **udp** の各キーワードは、このサービス オブジェクトが ICMP プロトコル、TCP プロトコル、または UDP プロトコルのいずれかのサービス オブジェクトであることを指定します。 *icmp-type* 引数は、ICMP タイプを指定します。 **icmp6** キーワードは、サービス タイプが ICMP バージョン 6 接続用であることを指定します。 *icmp6-type* 引数は、ICMP バージョン 6 タイプを指定します。 **source** キーワードは、送信元ポートを指定します。 **destination** キーワードは、宛先ポートを指定します。 *operator port* 引数は、プロトコルのポート設定をサポートする 1 つのポート / コードの値を指定します。TCP または UDP のポートの設定時には、「eq」、「neq」、「lt」、「gt」、および「range」を指定できます。「range」演算子を使用すると、開始ポートおよび終了ポートの一覧が表示されます。

例

サービス オブジェクトを作成するには、次のコマンドを入力します。

```
hostname (config)# object service SERVOBJECT1
hostname (config-service-object)# service tcp source eq www destination eq ssh
```

オブジェクト グループの設定

この項は、次の内容で構成されています。

プロトコル オブジェクト グループの追加

プロトコル オブジェクト グループを追加または変更するには、この項の手順を実行します。グループを追加した後、同じグループ名でこの手順を繰り返し、追加のオブジェクトを指定することで、必要に応じてさらにオブジェクトを追加できます。既存のオブジェクトを再入力する必要はありません。設定済みのコマンドは、コマンドの **no** 形式で削除するまで残ります。

手順の詳細

	コマンド	目的
ステップ 1	**object-group protocol** *obj_grp_id* **例：** `hostname(config)# object-group protocol tcp_udp_icmp`	プロトコル グループを追加します。*obj_grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） プロンプトがプロトコル コンフィギュレーション モードに変わります。
ステップ 2	**description** *text* **例：** `hostname(config-protocol)# description New Group`	（任意）説明を追加します。説明には、最大 200 文字を使用できます。
ステップ 3	**protocol-object** *protocol* **例：** `hostname(config-protocol)# protocol-object tcp`	グループでプロトコルを定義します。プロトコルごとにコマンドを入力します。protocol は、指定の IP プロトコルの数値識別子（1 ～ 254）またはキーワード識別子（たとえば、**icmp**、**tcp**、または **udp**）です。すべての IP プロトコルを含めるには、キーワード **ip** を使用します。指定できるプロトコルのリストについては、「プロトコルとアプリケーション」（P.B-5）を参照してください。

例

TCP、UDP、および ICMP のプロトコル グループを作成するには、次のコマンドを入力します。

```
hostname (config)# object-group protocol tcp_udp_icmp
```

```
hostname (config-protocol)# protocol-object tcp
hostname (config-protocol)# protocol-object udp
hostname (config-protocol)# protocol-object icmp
```

ネットワーク オブジェクト グループの追加

ネットワーク オブジェクト グループを追加または変更するには、この項の手順を実行します。グループを追加した後、同じグループ名でこの手順を繰り返し、追加のオブジェクトを指定することで、必要に応じてさらにオブジェクトを追加できます。既存のオブジェクトを再入力する必要はありません。設定済みのコマンドは、コマンドの no 形式で削除するまで残ります。

手順の詳細

	コマンド	目的
ステップ 1	**object-group network** *grp_id* **例：** hostname(config)# object-group network admins	ネットワーク グループを追加します。 *grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） プロンプトがプロトコル コンフィギュレーション モードに変わります。
ステップ 2	**description** *text* **例：** hostname(config-network)# Administrator Addresses	（任意）説明を追加します。説明には、最大 200 文字を使用できます。
ステップ 3	**network-object** {**object** *name* | host ip_address | ip_address mask} **例：** hostname(config-network)# network-object host 10.2.2.4	**object** キーワードは、ネットワーク オブジェクト グループに追加オブジェクトを追加します。 グループでネットワークを定義します。ネットワークまたはアドレスごとにコマンドを入力します。

例

3 人の管理者の IP アドレスを含むネットワーク グループを作成するには、次のコマンドを入力します。

```
hostname (config)# object-group network admins
hostname (config-protocol)# description Administrator Addresses
hostname (config-protocol)# network-object host 10.2.2.4
hostname (config-protocol)# network-object host 10.2.2.78
hostname (config-protocol)# network-object host 10.2.2.34
```

サービス オブジェクト グループの追加

サービス オブジェクト グループを追加または変更するには、この項の手順を実行します。グループを追加した後、同じグループ名でこの手順を繰り返し、追加のオブジェクトを指定することで、必要に応じてさらにオブジェクトを追加できます。既存のオブジェクトを再入力する必要はありません。設定済みのコマンドは、コマンドの **no** 形式で削除するまで残ります。

手順の詳細

	コマンド	目的
ステップ 1	**object-group service** *grp_id* {**tcp** | **udp** | **tcp-udp**} **例：** `hostname(config)# object-group service services1 tcp-udp`	サービス グループを追加します。 **object** キーワードは、サービス オブジェクト グループに追加オブジェクトを追加します。 *grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） **tcp**、**udp**、または **tcp-udp** のいずれかのキーワードで、追加するサービス（ポート）のプロトコルを指定します。DNS（ポート 53）のように、同じポート番号で TCP と UDP の両方を使用している場合は、**tcp-udp** キーワードを入力します。 プロンプトがサービス コンフィギュレーション モードに変わります。
ステップ 2	**description** *text* **例：** `hostname(config-service)# description DNS Group`	（任意）説明を追加します。説明には、最大 200 文字を使用できます。
ステップ 3	**port-object** {**eq** *port* | **range** *begin_port end_port*} **例：** `hostname(config-service)# port-object eq domain`	グループでポートを定義します。ポートまたはポート範囲ごとにコマンドを入力します。使用できるキーワードおよび予約済みポート割り当てのリストについては、「プロトコルとアプリケーション」（P.B-5）を参照してください。

例

DNS（TCP/UDP）、LDAP（TCP）、および RADIUS（UDP）が含まれたサービス グループを作成するには、次のコマンドを入力します。

```
hostname (config)# object-group service services1 tcp-udp
hostname (config-service)# description DNS Group
hostname (config-service)# port-object eq domain

hostname (config)# object-group service services2 udp
hostname (config-service)# description RADIUS Group
```

```
hostname (config-service)# port-object eq radius
hostname (config-service)# port-object eq radius-acct

hostname (config)# object-group service services3 tcp
hostname (config-service)# description LDAP Group
hostname (config-service)# port-object eq ldap
```

ICMP タイプ オブジェクト グループの追加

ICMP タイプ オブジェクト グループを追加または変更するには、この項の手順を実行します。グループを追加した後、同じグループ名でこの手順を繰り返し、追加のオブジェクトを指定することで、必要に応じてさらにオブジェクトを追加できます。既存のオブジェクトを再入力する必要はありません。設定済みのコマンドは、コマンドの **no** 形式で削除するまで残ります。

手順の詳細

	コマンド	目的
ステップ 1	**object-group icmp-type** *grp_id* **例：** `hostname(config)# object-group icmp-type ping`	ICMP タイプ オブジェクト グループを追加します。*grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.） プロンプトが ICMP タイプ コンフィギュレーション モードに変わります。
ステップ 2	**description** *text* **例：** `hostname(config-icmp-type)# description Ping Group`	（任意）説明を追加します。説明には、最大 200 文字を使用できます。
ステップ 3	**icmp-object** *icmp-type* **例：** `hostname(config-icmp-type)# icmp-object echo-reply`	ICMP タイプをグループで定義します。タイプごとにコマンドを入力します。ICMP タイプのリストについては、「ICMP タイプ」（P.B-9）を参照してください。

例

次のコマンドを入力して、echo-reply および echo（ping 制御に使用）が含まれる ICMP タイプ グループを作成します。

```
hostname (config)# object-group icmp-type ping
hostname (config-service)# description Ping Group
hostname (config-service)# icmp-object echo
hostname (config-service)# icmp-object echo-reply
```

オブジェクト グループのネスト

オブジェクト グループを階層型にネストして、1 つのオブジェクト グループが同じタイプの他のオブジェクト グループを含むようにできます。そして、ネストしたグループ オブジェクトと通常のオブジェクトは、単一のオブジェクト グループ内でさまざまに組み合わせることができます。

オブジェクト グループを同じタイプの別のオブジェクト グループ内にネストするには、まず、ネストするグループを作成し（「オブジェクト グループの設定」（P.8-5）を参照）、この項の手順を実行します。

手順の詳細

	コマンド	目的
ステップ 1	**object-group group** {{**protocol** | **network** | **icmp-type**} *grp_id* |**service** grp_id {**tcp** | **udp** | **tcp-udp**}} **例：** `hostname(config)# object-group network Engineering_group`	下位に別のオブジェクト グループをネストする指定のオブジェクト グループ タイプを追加または編集します。 **service**_*grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.）
ステップ 2	**group-object** *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	指定したグループをステップ 1 で指定したオブジェクト グループの下位に追加します。ネストするグループは、同じタイプである必要があります。ネストしたグループ オブジェクトと通常のオブジェクトは、単一のオブジェクト グループ内でさまざまに組み合わせることができます。

例

次のコマンドを入力して、さまざまな部門に所属する特権ユーザのネットワーク オブジェクト グループを作成します。

```
hostname (config)# object-group network eng
hostname (config-network)# network-object host 10.1.1.5
hostname (config-network)# network-object host 10.1.1.9
hostname (config-network)# network-object host 10.1.1.89

hostname (config)# object-group network hr
hostname (config-network)# network-object host 10.1.2.8
hostname (config-network)# network-object host 10.1.2.12

hostname (config)# object-group network finance
hostname (config-network)# network-object host 10.1.4.89
hostname (config-network)# network-object host 10.1.4.100
```

その後、3 つすべてのグループを次のようにネストします。

```
hostname (config)# object-group network admin
hostname (config-network)# group-object eng
hostname (config-network)# group-object hr
hostname (config-network)# group-object finance
```

ACE では次のように管理オブジェクト グループを指定するだけです。

```
hostname (config)# access-list ACL_IN extended permit ip object-group admin host
209.165.201.29
```

オブジェクト グループの削除

特定のオブジェクト グループまたは指定したタイプのすべてのオブジェクト グループを削除できますが、アクセス リストで使用されている場合は、そのオブジェクト グループを削除したり、空にすることはできません。

手順の詳細

ステップ 1　次のいずれかを実行します。

no object-group *grp_id* **例：** `hostname(config)# no object-group Engineering_host`	指定のオブジェクト グループを削除します。*grp_id* は最大長が 64 文字の文字列で、英字、数字、および次の文字を組み合わせることができます。 • 下線（_） • ダッシュ（-） • ピリオド（.）
clear object-group [**protocol** | **network** | **services** | **icmp-type**] **例：** `hostname(config)# clear-object group network`	指定したタイプのすべてのオブジェクト グループを削除します。 **（注）** タイプを入力しない場合、すべてのオブジェクト グループが削除されます。

オブジェクトとグループのモニタリング

オブジェクトおよびグループをモニタするには、次のコマンドを入力します。

コマンド	目的
show access-list	オブジェクトをグループ化せずに個々のエントリに拡張されるアクセス リスト エントリを表示します。
show running-config object-group	現在のすべてのオブジェクト グループを表示します。
show running-config object-group *grp_id*	現在のオブジェクト グループをグループ ID ごとに表示します。
show running-config object-group *grp_type*	現在のオブジェクト グループをグループ タイプごとに表示します。

正規表現の設定

正規表現は、ストリングそのものとしてテキスト ストリングと文字どおりに照合することも、*metacharacters* を使用してテキスト ストリングの複数のバリアントと照合することもできます。正規表現を使用して特定のアプリケーション トラフィックの内容と照合できます。たとえば、HTTP パケット内部の URL 文字列と照合できます。この項では、正規表現を作成する方法について説明します。次の項目で構成されています。

正規表現の作成

正規表現は、ストリングそのものとしてテキスト ストリングと文字どおりに照合することも、*metacharacters* を使用してテキスト ストリングの複数のバリアントと照合することもできます。正規表現を使用して特定のアプリケーション トラフィックの内容と照合できます。たとえば、HTTP パケット内部の URL 文字列と照合できます。

ガイドライン

Ctrl キーを押した状態で V キーを押すと、CLI において、疑問符（?）やタブなどの特殊文字をすべてエスケープできます。たとえば、コンフィギュレーションで **d?g** と入力するには、**d[Ctrl+V]?g** とキー入力します。

正規表現をパケットと照合する場合のパフォーマンスへの影響については、コマンド リファレンスの **regex** コマンドを参照してください。

（注） 最適化のために、ASA 1000V では、難読化解除された URL が検索されます。難読化解除では、複数のスラッシュ（/）が単一のスラッシュに圧縮されます。「http://」などの、一般的に 2 つのスラッシュが使用されるストリングでは、代わりに「http:/」を検索してください。

表 8-1 は、特殊な意味を持つメタ文字のリストです。

表 8-1　　regex メタ文字

文字	説明	注釈
.	ドット	任意の単一文字と一致します。たとえば、**d.g** は、dog、dag、dtg、およびこれらの文字を含む任意の単語（doggonnit など）に一致します。
(*exp***)**	サブ表現	サブ表現は、文字を周囲の文字から分離して、サブ表現に他のメタ文字を使用できるようにします。たとえば、**d(o|a)g** は dog および dag に一致しますが、**do|ag** は do および ag に一致します。また、サブ表現を繰り返し限定作用素とともに使用して、繰り返す文字を区別できます。たとえば、**ab(xy){3}z** は、abxyxyxyz に一致します。
|	代替	このメタ文字によって区切られている複数の表現のいずれかと一致します。たとえば、**dog|cat** は、dog または cat に一致します。
?	疑問符	直前の表現が 0 または 1 個存在することを示す修飾子。たとえば、**lo?se** は、lse または lose に一致します。 **（注）** Ctrl+V を入力してから疑問符を入力しないと、ヘルプ機能が呼び出されます。
*	アスタリスク	直前の表現が 0、1、または任意の個数存在することを示す修飾子。たとえば、**lo*se** は、lse、lose、loose などに一致します。

表 8-1　regex メタ文字 （続き）

文字	説明	注釈
+	プラス	直前の表現が少なくとも 1 個存在することを示す修飾子。たとえば、**lo+se** は、lose および loose に一致しますが、lse には一致しません。
{*x*} または {*x*,}	最小繰り返し限定作用素	少なくとも *x* 回繰り返します。たとえば、**ab(xy){2,}z** は、abxyxyz や abxyxyxyz などに一致します。
[*abc*]	文字クラス	カッコ内の任意の文字と一致します。たとえば、**[abc]** は、a、b、または c に一致します。
[^*abc*]	否定文字クラス	角カッコに含まれていない単一文字と一致します。たとえば、**[^abc]** は、a、b、c 以外の任意の文字に一致します。**[^A-Z]** は、大文字のアルファベット文字以外の任意の単一の文字に一致します。
[*a-c*]	文字範囲クラス	範囲内の任意の文字と一致します。**[a-z]** は、任意の小文字のアルファベット文字に一致します。文字と範囲を組み合わせて使用することもできます。**[abcq-z]** および **[a-cq-z]** は、a、b、c、q、r、s、t、u、v、w、x、y、z に一致します。 ダッシュ（-）文字は、角カッコ内の最初の文字または最後の文字である場合にのみリテラルとなります（**[abc-]** や **[-abc]**）。
""	引用符	文字列の末尾または先頭のスペースを保持します。たとえば、**" test"** では一致を探すときに先頭のスペースを保持します。
^	キャレット	行の先頭を指定します。
\	エスケープ文字	メタ文字とともに使用すると、リテラル文字と一致します。たとえば、**\[** は左角カッコに一致します。
char	文字	文字がメタ文字でない場合は、リテラル文字と一致します。
\r	復帰	復帰 0x0d と一致します。
\n	改行	改行 0x0a と一致します。
\t	Tab	タブ 0x09 と一致します。
\f	改ページ	フォーム フィード 0x0c と一致します。
\x*NN*	エスケープされた 16 進数	16 進数（厳密に 2 桁）を使用した ASCII 文字と一致します。
NNN	エスケープされた 8 進数	8 進数（厳密に 3 桁）としての ASCII 文字と一致します。たとえば、文字 040 はスペースを表します。

手順の詳細

ステップ 1　正規表現をテストして、一致するはずの対象と一致することを確認するには、次のコマンドを入力します。

```
hostname(config)# test regex input_text regular_expression
```

input_text 引数は、正規表現を使用して照合する、長さが最大で 201 文字の文字列です。

regular_expression 引数の長さは、最大 100 文字です。

Ctrl+V を使用して、CLI の特殊文字をすべてエスケープします。たとえば、**test regex** コマンドの入力テキストにタブを入力するには、**test regex "test[Ctrl+V Tab]" "test\t"** と入力する必要があります。

正規表現が入力テキストと一致する場合は、次のメッセージが表示されます。

```
INFO: Regular expression match succeeded.
```

正規表現が入力テキストと一致しない場合は、次のメッセージが表示されます。

```
INFO: Regular expression match failed.
```

ステップ 2　テスト後に正規表現を追加するには、次のコマンドを入力します。

```
hostname(config)# regex name regular_expression
```

name 引数の長さは、最大 40 文字です。

regular_expression 引数の長さは、最大 100 文字です。

例

次に、インスペクション ポリシー マップで使用する 2 つの正規表現を作成する例を示します。

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
```

正規表現クラス マップの作成

正規表現クラス マップで、1 つ以上の正規表現を指定します。正規表現クラス マップを使用して、特定のトラフィックの内容を照合できます。たとえば、HTTP パケット内の URL 文字列の照合が可能です。

手順の詳細

ステップ 1　「正規表現の設定」の項の説明に従って、正規表現を 1 つ以上作成します。

ステップ 2　次のコマンドを入力して、クラス マップを作成します。

```
hostname(config)# class-map type regex match-any class_map_name
hostname(config-cmap)#
```

class_map_name は、最大 40 文字の文字列です。「class-default」という名前は予約されています。すべてのタイプのクラス マップで同じ名前スペースが使用されるため、別のタイプのクラス マップですでに使用されている名前は再度使用できません。

match-any キーワードにより、トラフィックが少なくとも 1 つの正規表現と一致する場合には、そのトラフィックがクラス マップと一致するように指定します。

CLI はクラスマップ コンフィギュレーション モードに移行します。

ステップ 3　（任意）次のコマンドを入力して、クラス マップの説明を追加します。

```
hostname(config-cmap)# description string
```

ステップ 4　正規表現ごとに次のコマンドを入力して、クラス マップに含める正規表現を指定します。

```
hostname(config-cmap)# match regex regex_name
```

例

次に、2 つの正規表現を作成し、これを正規表現クラス マップに追加する例を示します。トラフィックに「example.com」または「example2.com」というストリングが含まれている場合、このトラフィックはクラス マップと一致しています。

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2
```

拡張アクセス リストのアクティベーションのスケジュール設定

この項は、次の内容で構成されています。

アクセス リストのアクティベーションのスケジュール設定に関する情報

ACE に時間範囲を適用することで、アクセス リストの各 ACE が、1 日および週の特定の時刻にアクティブになるようにスケジュールを設定できます。

アクセス リストのアクティベーションのスケジュール設定におけるガイドラインと制限事項

アクセス リストでオブジェクト グループを使用する際には、次のガイドラインと制限事項が適用されます。

- ACL を非アクティブにするための指定の終了時刻の後、約 80 ～ 100 秒の遅延が発生する場合があります。たとえば、指定の終了時刻が 3:50 の場合、この 3:50 は終了時刻に含まれているため、コマンドは、3:51:00 ～ 3:51:59 の間に呼び出されます。コマンドが呼び出された後、ASA 1000V は現在実行されているすべてのタスクを終了し、コマンドに ACL を無効にさせます。
- **time-range** コマンドごとに、複数の定期的なエントリが許可されます。**time-range** コマンドに **absolute** 値と **periodic** 値の両方が指定されている場合、**periodic** コマンドは **absolute** の開始時刻になって初めて評価され、**absolute** の終了時刻に達した後は評価されません。

時間範囲の設定と適用

時間範囲を追加して時間ベースのアクセス リストを実装できます。時間範囲を特定するには、この項の手順を実行します。

手順の詳細

	コマンド	目的
ステップ 1	**time-range** *name* **例：** `hostname(config)# time range Sales`	時間範囲の名前を特定します。
ステップ 2	次のいずれかを実行します。	
	periodic *days-of-the-week time* **to** [*days-of-the-week*] *time* **Example:** `hostname(config-time-range)# periodic monday 7:59 to friday 17:01`	定期的な時間範囲を指定します。 *days-of-the-week* には次の値を指定できます。 • **monday**、**tuesday**、**wednesday**、**thursday**、**friday**、**saturday**、または **sunday**。 • **daily** • **weekdays** • **weekend** *time* の形式は、*hh*:*mm* です。たとえば、8:00 は午前 8 時です。午後 8 時は 20:00 と指定します。
	absolute **start** *time date* [**end** *time date*] **例：** `hostname(config-time-range)# absolute start 7:59 2 january 2009`	絶対的な時間範囲を指定します。 *time* の形式は、*hh*:*mm* です。たとえば、8:00 は午前 8 時です。午後 8 時は 20:00 と指定します。 *date* の形式は、*day month year* です。たとえば、**1 january 2006** と指定します。
ステップ 3	**access-list** *access_list_name* [**extended**] *{deny | permit}...*[**time-range** *name*] **例：** `hostname(config)# access list Marketing extended deny tcp host 209.165.200.225 host 209.165 201.1 time-range Pacific_Coast`	ACE へ時間範囲を適用します。 **（注）** ACE のロギングもイネーブルにするには、**log** キーワードを **time-range** キーワードの前に使用します。**inactive** キーワードを使用して ACE をディセーブルにする場合は、**inactive** キーワードを最後のキーワードとして使用します。 **access-list** コマンドの完全な構文については、第 9 章「拡張アクセス リストの追加」を参照してください。

例

次に、「Sales」という名前のアクセス リストを「New_York_Minute」という名前の時間範囲にバインドする例を示します。

```
hostname(config)# access-list Sales line 1 extended deny tcp host 209.165.200.225 host
209.165.201.1 time-range New_York_Minute
```

アクセス リストのアクティベーションのスケジュール設定例

次に、2006 年 1 月 1 日の午前 8 時に始まる絶対的な時間範囲の例を示します。終了時刻も終了日も指定されていないため、時間範囲は事実上無期限になります。

```
hostname(config)# time-range for2006
hostname(config-time-range)# absolute start 8:00 1 january 2006
```

次に、平日の午前 8 時～午後 6 時に毎週繰り返される定期的な時間範囲の例を示します。

```
hostname(config)# time-range workinghours
hostname(config-time-range)# periodic weekdays 8:00 to 18:00
```