オブジェクトの設定

この章では、コンフィギュレーションで使用するための、再利用可能な名前付きオブジェクトおよびグループを設定する方法について説明します。次の項が含まれます。

オブジェクトに関する情報

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、ASA コンフィギュレーションの中で定義して、インライン IP アドレス、サービス、名前などの代わりに使用できます。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。

オブジェクトのライセンス要件

モデル	ライセンス要件
すべてのモデル	基本ライセンス

注意事項と制限事項

コンテキスト モードのガイドライン

シングル コンテキスト モードとマルチ コンテキスト モードでサポートされています。

ファイアウォール モードのガイドライン

ルーテッド ファイアウォール モードとトランスペアレント ファイアウォール モードでサポートされています。

IPv6 のガイドライン

- IPv6 をサポートします。
- ASA は、ネストされた IPv6 ネットワーク オブジェクト グループはサポートしません。したがって、IPv6 エントリが含まれるオブジェクトを別の IPv6 オブジェクト グループの下でグループ化することはできません。
- 1 つのネットワーク オブジェクト グループの中で IPv4 および IPv6 のエントリを混在させることができます。NAT に対しては、混合オブジェクト グループは使用できません。

その他のガイドラインと制限事項

- オブジェクトには、一意の名前を付ける必要があります。「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。
- 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。
- コマンドで使用されているオブジェクトを削除したり、空にしたりすることはできません。

オブジェクトの設定

ネットワーク オブジェクトとグループの設定

この項では、ネットワーク オブジェクトおよびグループの設定方法について説明します。次の項目を取り上げます。

ネットワーク オブジェクトの設定

1 つのネットワーク オブジェクトには、1 つのホスト、ネットワーク IP アドレス、または IP アドレス範囲、完全修飾ドメイン名（FQDN）を入れることができます。また、オブジェクトに対して NAT ルールをイネーブルにすることもできます（FQDN オブジェクトを除く）。（詳細については、ファイアウォール コンフィギュレーション ガイドの Chapter 4, “Configuring Network Object NAT,” 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* | **fqdn** *fully_qualified_domain_name*} **例：** `hostname(config-network-object)# host 10.2.2.2`	名前付きオブジェクトに IP アドレスまたは FQDN を割り当てます。 **（注）** FQDN オブジェクトに対して NAT を設定することはできません。
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
```

ネットワーク オブジェクト グループの設定

ネットワーク オブジェクト グループには、インライン ネットワークと同様に複数のネットワーク オブジェクトを入れることができます。ネットワーク オブジェクト グループは、IPv4 と IPv6 の両方のアドレスの混在をサポートできます。

制約事項

IPv4 と IPv6 が混在するオブジェクト グループや、FQDN オブジェクトが含まれているオブジェクト グループを、NAT に使用することはできません。

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	次のグループ メンバの 1 つ以上を追加します。	
	network-object object *name* **例：** `hostname(config-network)# network-object host 10.2.2.4`	オブジェクトをネットワーク オブジェクト グループに追加します。
	network-object {**host** *ipv4_address* | *ipv4_address mask* | *ipv6-address/prefix-length*} **例：** `hostname(config-network)# network-object host 10.2.2.4`	ホストまたはネットワーク（IPv4 または IPv6）をインラインで追加します。
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。

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
```

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
```

サービス オブジェクトとサービス グループの設定

サービス オブジェクトとグループでは、プロトコルおよびポートを指定します。ここでは、サービス オブジェクト、サービス グループ、TCP と UDP のポート サービス グループ、プロトコル グループ、および ICMP グループを設定する方法を説明します。説明する項目は次のとおりです。

サービス オブジェクトの設定

サービス オブジェクトは、プロトコル、ICMP、ICMPv6、TCP、または UDP のポートあるいはポート範囲を含むことができます。

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* [*icmp_code*] | **icmp6** *icmp6-type* [*icmp_code*] | {**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。任意の *icmp_code* では、1 ～ 255 の ICMP コードを指定します。 **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。任意の *icmp_code* では、1 ～ 255 の ICMP コードを指定します。 TCP または UDP の場合は、**source** キーワードで送信元ポートを指定します。 TCP または UDP の場合は、**destination** キーワードで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
```

サービス グループの設定

1 つのサービス オブジェクト グループには、さまざまなプロトコルが混在しています。必要に応じて、TCP または UDP の送信元および宛先のポートも入れることができます。

手順の詳細

	コマンド	目的
ステップ 1	**object-group service** *grp_id* **例：** `hostname(config)# object-group service services1`	サービス 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	次のグループ メンバの 1 つ以上を追加します。	
	service-object *protocol* **例：** `hostname(config-service)# service-object ipsec`	プロトコル名または番号（0 ～ 255）を指定します。
	service-object {**tcp** | **udp** | **tcp-udp**} [**source** *operator number*] [**destination** *operator number*] **例：** `hostname(config-service)# port-object eq domain`	送信元と宛先の一方または両方のポート（0 ～ 65535）を指定できます。サポートされる名前のリストについては、CLI ヘルプを参照してください。有効な演算子は、次のとおりです。 • eq：ポート番号に等しい。 • gt：ポート番号より大きい。 • lt：ポート番号より小さい。 • neq：ポート番号と等しくない。 • range：ポート範囲。2 つの番号をスペースで区切って指定します。たとえば、「range 1024 4500」です。
	service-object {**icmp** [*icmp_type* [*icmp_code*]] | **icmp6** [*icmp6_type* [*icmp_code*]]} **例：** `hostname(config-service)# port-object eq domain`	サービス タイプが ICMP または ICMPv6 接続用であることを指定します。任意で ICMP タイプを名前または番号（0 ～ 255）で指定できます。 任意の *icmp_code* では、1 ～ 255 の ICMP コードを指定します。
	service-object object *name* **例：** `hostname(config-service)# port-object eq domain`	サービス オブジェクト名を指定します。これは、**object service** コマンドで作成されたものです。

	コマンド	目的
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

例

次の例では、TCP と UDP の両方のサービスを同じサービス オブジェクト グループに追加する方法を示します。

```
hostname(config)# object-group service CommonApps
hostname(config-service-object-group)# service-object destination tcp eq ftp
hostname(config-service-object-group)# service-object destination tcp-udp eq www
hostname(config-service-object-group)# service-object destination tcp eq h323
hostname(config-service-object-group)# service-object destination tcp eq https
hostname(config-service-object-group)# service-object destination udp eq ntp
```

次の例では、複数のサービス オブジェクトを同じサービス オブジェクト グループに追加する方法を示します。

```
hostname(config)# service object SSH
hostname(config-service-object)# service tcp destination eq ssh
hostname(config)# service object EIGRP
hostname(config-service-object)# service eigrp
hostname(config)# service object HTTPS
hostname(config-service-object)# service tcp source range 0 1024 destination eq https
hostname(config)# object-group service Group1
hostname(config-service-object-group)# service-object object SSH
hostname(config-service-object-group)# service-object object EIGRP
hostname(config-service-object-group)# service-object object HTTPS
```

TCP または UDP ポート サービス グループの設定

TCP または UDP サービス グループには、特定のプロトコル（TCP、UDP、または TCP-UDP）のポートのグループが含まれます。

	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（.） 追加するサービス（ポート）のプロトコルを、**tcp**、**udp**、または **tcp-udp** キーワードで指定します。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	次のグループ メンバの 1 つ以上を追加します。	
	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.44-11）を参照してください。
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

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 グループの設定

1 つの ICMP グループに、複数の ICMP タイプが含まれます。

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	次のグループ メンバの 1 つ以上を追加します。	
	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.44-16）を参照してください。
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

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
```

プロトコル グループの設定

1 つのプロトコル グループに、複数の IP プロトコル タイプが含まれます。

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	次のグループ メンバの 1 つ以上を追加します。	
	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.44-11）を参照してください。
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

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
```

ローカル ユーザ グループの設定

作成したローカル ユーザ グループは、アイデンティティ ファイアウォール（IDFW）をサポートする機能で使用できます。そのグループを拡張 ACL に入れると、たとえばアクセス ルールでも使用できるようになります。

ASA は、Active Directory ドメイン コントローラでグローバルに定義されているユーザ グループについて、Active Directory サーバに LDAP クエリーを送信します。ASA は、そのグループをアイデンティティ ベースのルール用にインポートします。ただし、ローカライズされたセキュリティ ポリシーを持つローカル ユーザ グループを必要とする、グローバルに定義されていないネットワーク リソースが ASA によりローカライズされている場合があります。ローカル ユーザ グループには、Active Directory からインポートされる、ネストされたグループおよびユーザ グループを含めることができます。ASA は、ローカル グループおよび Active Directory グループを統合します。

ユーザは、ローカル ユーザ グループと Active Directory からインポートされたユーザ グループに属することができます。

前提条件

IDFW をイネーブルにするには、第 33 章「アイデンティティ ファイアウォールの設定」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-group user** *user_group_name* **例：** `hostname(config)# object-group user users1`	アイデンティティ ファイアウォールでアクセスを制御するために使用できるオブジェクト グループを定義します。
ステップ 2	次のグループ メンバの 1 つ以上を追加します。	
	user *domain_NetBIOS_name\user_name* **例：** `hostname(config-user-object-group)# user SAMPLE\users1`	アクセス ルールに追加するユーザを指定します。 *user_name* には、[a-z]、[A-Z]、[0-9]、[!@#$%^&()-_{}.] など、あらゆる文字を使用できます。*domain_NetBIOS_name\user_name* にスペースを含める場合は、ドメイン名とユーザ名を引用符で囲む必要があります。 *user_name* には、LOCAL ドメインの一部または ASA が Active Directory ドメインからインポートしたユーザを指定できます。 *domain_NetBIOS_name* が AAA サーバと関連付けられている場合は、*user_name* を一意ではない場合がある Common Name（CN; 通常名）ではなく Active Directory の一意な sAMAccountName とする必要があります。 *domain_NetBIOS_name* には、LOCAL を指定することも、**user-identity domain** *domain_NetBIOS_name* **aaa-server** *aaa_server_group_tag* コマンドで指定された実際のドメイン名を指定することもできます。

	コマンド	目的
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

セキュリティ グループ オブジェクト グループの設定

作成したセキュリティ グループ オブジェクト グループは、Cisco TrustSec をサポートする機能で使用できます。そのグループを拡張 ACL に入れると、たとえばアクセス ルールで使用できるようになります。

Cisco TrustSec と統合されているときは、ASA は ISE からセキュリティ グループの情報をダウンロードします。ISE はアイデンティティ リポジトリとしても動作し、Cisco TrustSec タグからユーザ アイデンティティへのマッピングと、Cisco TrustSec タグからサーバ リソースへのマッピングを行います。セキュリティ グループ アクセス リストのプロビジョニングおよび管理は、中央集中型で ISE 上で行います。

ただし、ASA には、グローバルには定義されていない、ローカライズされたネットワーク リソースが存在することがあり、そのようなリソースにはローカル セキュリティ グループとローカライズされたセキュリティ ポリシーが必要です。ローカル セキュリティ グループには、ISE からダウンロードされた、ネストされたセキュリティ グループを含めることができます。ASA は、ローカルと中央のセキュリティ グループを統合します。

ASA 上でローカル セキュリティ グループを作成するには、ローカル セキュリティ オブジェクト グループを作成します。1 つのローカル セキュリティ オブジェクト グループに、1 つ以上のネストされたセキュリティ オブジェクト グループまたはセキュリティ ID またはセキュリティ グループ名を入れることができます。ユーザは、ASA 上に存在しない新しいセキュリティ ID またはセキュリティ グループ名を作成することもできます。

ASA 上で作成したセキュリティ オブジェクト グループは、ネットワーク リソースへのアクセスの制御に使用できます。セキュリティ オブジェクト グループを、アクセス グループやサービス ポリシーの一部として使用できます。

前提条件

TrustSec をイネーブルにするには、第 34 章「Cisco TrustSec と統合するための ASA 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-group security** *objgrp_name* **例：** `hostname(config)# object-group security mktg-sg`	セキュリティ グループ オブジェクトを作成します。 *objgrp_name* はグループの名前です。32 バイトの文字列を、大文字と小文字を区別して入力します。 *objgrp_name* には、[a-z]、[A-Z]、[0-9]、[!@#$%^&()-_{}.] など、あらゆる文字を使用できます。
ステップ 2	次のグループ メンバの 1 つ以上を追加します。	
	security-group {**tag** *sgt#* | **name** *sg_name*} **例：** `hostname(config)# security-group name mktg`	セキュリティ グループ オブジェクトのタイプを、インライン タグまたは名前付きオブジェクトとして指定します。 • **tag** *sgt#*：セキュリティ タイプが「タグ」の場合は、1 ～ 65533 の番号を入力します。 • **name** *sg_name*：セキュリティ タイプが「名前」の場合は、32 バイトの文字列を、大文字と小文字を区別して入力します。*sg_name* には、[a-z]、[A-Z]、[0-9]、[!@#$%^&()-_{}.] など、あらゆる文字を使用できます。 SGT は、ISE による IEEE 802.1X 認証、Web 認証、または MAC 認証バイパス（MAB）を通してデバイスに割り当てられます。セキュリティ グループの名前は ISE 上で作成され、セキュリティ グループをわかりやすい名前で識別できるようになります。セキュリティ グループ テーブルによって、SGT がセキュリティ グループ名にマッピングされます。
	group-object *group_id* **例：** `hostname(config-network)# group-object Engineering_groups`	既存のオブジェクト グループをこのオブジェクト グループの下に追加します。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。
ステップ 3	**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。

例

次の例では、セキュリティ グループ オブジェクトを設定する方法を示します。

```
hostname(config)# object-group security mktg-sg
hostname(config)# security-group name mktg
hostname(config)# security-group tag 1
```

次の例では、セキュリティ グループ オブジェクトを設定する方法を示します。

```
hostname(config)# object-group security mktg-sg-all
hostname(config)# security-group name mktg-managers
hostname(config)# group-object mktg-sg // nested object-group
```

正規表現の設定

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 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 17-1 は、特殊な意味を持つメタ文字のリストです。

表 17-1　　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。

表 17-1　regex メタ文字 （続き）

文字	説明	注釈
*	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。
+	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。

前提条件

「正規表現の作成」（P.17-15）の説明に従って、正規表現を 1 つ以上作成します。

手順の詳細

ステップ 1　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 2　（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 3　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
```

時間範囲の設定

再利用可能コンポーネントを作成して、その中で開始と終了の時間を定義しておき、さまざまなセキュリティ機能に適用することができます。時間範囲を 1 回だけ定義すれば、後は時間範囲を選択して、スケジューリングが必要なさまざまなオプションに適用できます。

時間範囲機能を使用して時間の範囲を定義し、トラフィックのルールやアクションに使用できます。たとえば、アクセス リストに時間範囲を設定すると、ASA のアクセスを制限できます。

時間範囲は、開始時間、終了時間、およびオプションの繰り返しエントリで構成されます。

ガイドライン

- 1 つの時間範囲に対して、複数の periodic エントリを指定できます。1 つの時間範囲に absolute 値と periodic 値の両方が指定されている場合は、periodic 値は absolute の開始時刻に到達した後にのみ評価され、absolute の終了時刻に到達した後は評価されません。
- 時間範囲を作成してもデバイスへのアクセスは制限されません。この手順では、時間範囲だけを定義します。

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。	

コマンド	目的
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。

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
```

オブジェクトのモニタリング

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。

オブジェクトの機能履歴

表 17-2 に、各機能変更と、それが実装されたプラットフォーム リリースを示します。

表 17-2　オブジェクト グループの機能履歴

機能名	プラットフォーム リリース	機能情報
オブジェクト グループ	7.0(1)	オブジェクト グループにより、アクセス リストの作成とメンテナンスが簡略化されます。 **object-group** *protocol*、**object-group** *network*、**object-group** *service*、**object-group** *icmp_type* コマンドを導入または変更しました。
正規表現およびポリシー マップ	7.2(1)	インスペクション ポリシー マップで使用される正規表現およびポリシー マップが導入されました。**class-map type regex** コマンド、**regex** コマンド、および **match regex** コマンドが導入されました。
オブジェクト	8.3(1)	オブジェクトのサポートが導入されました。 **object-network**、**object-service**、**object-group** *network*、**object-group** *service*、**network object**、**access-list extended**、**access-list webtype**、**access-list remark** の各コマンドが導入または変更されました。
アイデンティティ ファイアウォールでのユーザ オブジェクト グループの使用	8.4(2)	アイデンティティ ファイアウォールのためのユーザ オブジェクト グループが導入されました。 **object-network user**、**user** の各コマンドが導入されました。
IPv4 および IPv6 の混合ネットワーク オブジェクト グループ	9.0(1)	以前は、ネットワーク オブジェクト グループに含まれているのは、すべて IPv4 アドレスであるか、すべて IPv6 アドレスでなければなりませんでした。現在では、ネットワーク オブジェクト グループが、IPv4 と IPv6 の両方のアドレスの混合をサポートするようになりました。 **（注）**　混合オブジェクト グループを NAT に使用することはできません。 **object-group network** コマンドが変更されました。
Cisco TrustSec のためのセキュリティ グループ オブジェクト グループ	8.4(2)	TrustSec のためのセキュリティ グループ オブジェクト グループが導入されました。 **object-network security**、**security** の各コマンドが導入されました。
ICMP コードによって ICMP トラフィックをフィルタリングするための拡張 ACL とオブジェクト機能拡張	9.0(1)	ICMP コードに基づいて ICMP トラフィックの許可または拒否ができるようになりました。 **access-list extended**、**service-object**、**service** の各コマンドが導入または変更されました。