Objects for Access Control

Objects are reusable components for use in your configuration. You can define and use them in Cisco ASA configurations in the place of inline IP addresses, services, names, and so on. Objects make it easy to maintain your configurations because you can modify an object in one place and have it be reflected in all other places that are referencing it. Without objects you would have to modify the parameters for every feature when required, instead of just once. For example, if a network object defines an IP address and subnet mask, and you want to change the address, you only need to change it in the object definition, not in every feature that refers to that IP address.

Guidelines for Objects

IPv6 Guidelines

Supports IPv6 with the following restrictions:

- You can mix IPv4 and IPv6 entries in a network object group, but you cannot use a mixed object group for NAT.

Additional Guidelines and Limitations

- Objects must have unique names, because objects and object groups share the same name space. While you might want to create a network object group named “Engineering” and a service object group named “Engineering,” you need to add an identifier (or “tag”) to the end of at least one object group name to make it unique. For example, you can use the names “Engineering_admins” and “Engineering_hosts” to make the object group names unique and to aid in identification.
- Object names are limited to 64 characters, including letters, numbers, and these characters: .!@#$%^&()-_{}. Object names are case-sensitive.
- You cannot remove an object or make an object empty if it is used in a command, unless you enable forward referencing (the **forward-reference enable** command).

Configure Objects

The following sections describe how to configure objects that are primarily used on access control.

Configure Network Objects and Groups

Network objects and groups identify IP addresses or host names. Use these objects in access control lists to simplify your rules.

Configure a Network Object

A network object can contain a host, a network IP address, a range of IP addresses, or a fully qualified domain name (FQDN).

You can also enable NAT rules on the object (excepting FQDN objects). For more information about configuring object NAT, see Network Address Translation (NAT).

Procedure

Step 1 Create or edit a network object using the object name: **object network** *object_name*

Example:

```
hostname(config)# object network email-server
```

Step 2 Add an address to the object using one of the following commands. Use the **no** form of the command to remove the object.

- **host** {*IPv4_address* | *IPv6_address*}—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8::0DB8:800:200C:417A.
- **subnet** {*IPv4_address IPv4_mask* | *IPv6_address/IPv6_prefix*}—The address of a network. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8:0:CD30::/60.
- **range** *start_address end_address*—A range of addresses. You can specify IPv4 or IPv6 ranges. Do not include masks or prefixes.
- **fqdn** [**v4** | **v6**] *fully_qualified_domain_name*—A fully-qualified domain name, that is, the name of a host, such as www.example.com. Specify **v4** to limit the address to IPv4, and **v6** for IPv6. If you do not specify an address type, IPv4 is assumed.

Example:

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

Step 3 (Optional) Add a description: **description** *string*

Configure a Network Object Group

Network object groups can contain multiple network objects as well as inline networks or hosts. Network object groups can include a mix of both IPv4 and IPv6 addresses.

However, you cannot use a mixed IPv4 and IPv6 object group for NAT, or object groups that include FQDN objects.

Procedure

Step 1 Create or edit a network object group using the object name: **object-group network** *group_name*

Example:

```
hostname(config)# object-group network admin
```

Step 2 Add objects and addresses to the network object group using one or more of the following commands. Use the **no** form of the command to remove an object.

- **network-object host** {*IPv4_address* | *IPv6_address*}—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8::0DB8:800:200C:417A.
- **network-object** {*IPv4_address IPv4_mask* | *IPv6_address/IPv6_prefix*}—The address of a network or host. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8:0:CD30::/60.
- **network-object object** *object_name*—The name of an existing network object.
- **group-object** *object_group_name*—The name of an existing network object group.

Example:

```
hostname(config-network-object-group)# network-object 10.1.1.0 255.255.255.0
hostname(config-network-object-group)# network-object 2001:db8:0:cd30::/60
hostname(config-network-object-group)# network-object host 10.1.1.1
hostname(config-network-object-group)# network-object host 2001:DB8::0DB8:800:200C:417A
hostname(config-network-object-group)# network-object object existing-object-1
hostname(config-network-object-group)# group-object existing-network-object-group
```

Step 3 (Optional) Add a description: **description** *string*

Examples

To create a network group that includes the IP addresses of three administrators, enter the following commands:

```
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
```

Create network object groups for privileged users from various departments by entering the following commands:

```
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
```

You then nest all three groups together as follows:

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

Configure Service Objects and Service Groups

Service objects and groups identify protocols and ports. Use these objects in access control lists to simplify your rules.

Configure a Service Object

A service object can contain a single protocol specification.

Procedure

Step 1 Create or edit a service object using the object name: **object service** *object_name*

Example:

```
hostname(config)# object service web
```

Step 2 Add a service to the object using one of the following commands. Use the **no** form of the command to remove an object.

- **service** *protocol*—The name or number (0-255) of an IP protocol. Specify **ip** to apply to all protocols.
- **service** {**icmp** | **icmp6**} [*icmp-type* [*icmp_code*]]—For ICMP or ICMP version 6 messages. You can optionally specify the ICMP type by name or number (0-255) to limit the object to that message type. If you specify a type, you can optionally specify an ICMP code for that type (1-255). If you do not specify the code, then all codes are used.

- **service** {**tcp** | **udp** | **sctp**} [**source** *operator port*] [**destination** *operator port*]—For TCP, UDP or SCTP. You can optionally specify ports for the source, destination, or both. You can specify the port by name or number. The operator can be one of the following:
  - **lt**—less than.
  - **gt**—greater than.
  - **eq**—equal to.
  - **neq**—not equal to.
  - **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example, **range 100 200**.

Example:

```
hostname(config-service-object)# service tcp destination eq http
```

Step 3 (Optional) Add a description: **description** *string*

Configure a Service Group

A service object group includes a mix of protocols, if desired, including optional source and destination ports for protocols that use them, and ICMP type and code.

Before you begin

You can model all services using the generic service object group, which is explained here. However, you can still configure the types of service group objects that were available prior to ASA 8.3(1). These legacy objects include TCP/UDP/TCP-UDP port groups, protocol groups, and ICMP groups. The contents of these groups are equivalent to the associated configuration in the generic service object group, with the exception of ICMP groups, which do not support ICMP6 or ICMP codes. If you still want to use these legacy objects, for detailed instructions, see the **object-service** command description in the command reference on Cisco.com.

Procedure

Step 1 Create or edit a service object group using the object name: **object-group service** *object_name*

Example:

```
hostname(config)# object-group service general-services
```

Step 2 Add objects and services to the service object group using one or more of the following commands. Use the **no** form of the command to remove an object.

- **service-object** *protocol*—The name or number (0-255) of an IP protocol. Specify **ip** to apply to all protocols.
- **service-object** {**icmp** | **icmp6**} [*icmp-type* [*icmp_code*]]—For ICMP or ICMP version 6 messages. You can optionally specify the ICMP type by name or number (0-255) to limit the object to that message

type. If you specify a type, you can optionally specify an ICMP code for that type (1-255). If you do not specify the code, then all codes are used.

- **service-object** {**tcp** | **udp** | **tcp-udp** | **sctp**} [**source** *operator port*] [**destination** *operator port*]—For TCP, UDP, or both, or for SCTP. You can optionally specify ports for the source, destination, or both. You can specify the port by name or number. The operator can be one of the following:
  - **lt**—less than.
  - **gt**—greater than.
  - **eq**—equal to.
  - **neq**—not equal to.
  - **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example, **range 100 200**.
- **service-object object** *object_name*—The name of an existing service object.
- **group-object** *object_group_name*—The name of an existing service object group.

Example:

```
hostname(config-service-object-group)# service-object ipsec
hostname(config-service-object-group)# service-object tcp destination eq domain
hostname(config-service-object-group)# service-object icmp echo
hostname(config-service-object-group)# service-object object my-service
hostname(config-service-object-group)# group-object Engineering_groups
```

Step 3 (Optional) Add a description: **description** *string*

Examples

The following example shows how to add both TCP and UDP services to a service object group:

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

The following example shows how to add multiple service objects to a service object group:

```
hostname(config)# object service SSH
hostname(config-service-object)# service tcp destination eq ssh
hostname(config)# object service EIGRP
hostname(config-service-object)# service eigrp
hostname(config)# object service HTTPS
hostname(config-service-object)# service tcp source range 1 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
```

Configure Local User Groups

You can create local user groups for use in features that support the identity firewall by including the group in an extended ACL, which in turn can be used in an access rule, for example.

The ASA sends an LDAP query to the Active Directory server for user groups globally defined in the Active Directory domain controller. The ASA imports these groups for identity-based rules. However, the ASA might have localized network resources that are not defined globally that require local user groups with localized security policies. Local user groups can contain nested groups and user groups that are imported from Active Directory. The ASA consolidates local and Active Directory groups.

A user can belong to local user groups and user groups imported from Active Directory.

Because you can use usernames and user group names directly in an ACL, you need to configure local user groups only if:

- You want to create a group of users defined in the LOCAL database.
- You want to create a group of users or user groups that are not captured in a single user group defined on the AD server.

Procedure

Step 1 Create or edit a user object group using the object name: **object-group user** *group_name*

Example:

```
hostname(config)# object-group user admins
```

Step 2 Add users and groups to the user object group using one or more of the following commands. Use the **no** form of the command to remove an object.

- **user** [*domain_NETBIOS_name*\]*username*—A username. If there is a space in the domain name or username, you must enclose the domain name and user name in quotation marks. The domain name can be LOCAL (for users defined in the local database) or an Active Directory (AD) domain name as specified in the **user-identity domain** *domain_NetBIOS_name* **aaa-server** *aaa_server_group_tag* command. When adding users defined in an AD domain, the *user_name* must be the Active Directory sAMAccountName, which is unique, instead of the common name (cn), which might not be unique. If you do not specify a domain name, the default is used, which is either LOCAL or the one defined on the **user-identity default-domain** command.
- **user-group** [*domain_NETBIOS_name*\\]*username*—A user group. If there is a space in the domain name or group name, you must enclose the domain name and group name in quotation marks. Note the double \\ that separates the domain and group names.
- **group-object** *object_group_name*—The name of an existing user object group.

Example:

```
hostname(config-user-object-group)# user EXAMPLE\admin
hostname(config-user-object-group)# user-group EXAMPLE\\managers
hostname(config-user-object-group)# group-object local-admins
```

Step 3 (Optional) Add a description: **description** *string*

Configure Security Group Object Groups

You can create security group object groups for use in features that support Cisco TrustSec by including the group in an extended ACL, which in turn can be used in an access rule, for example.

When integrated with Cisco TrustSec, the ASA downloads security group information from the ISE. The ISE acts as an identity repository, by providing Cisco TrustSec tag-to-user identity mapping and Cisco TrustSec tag-to-server resource mapping. You provision and manage security group ACLs centrally on the ISE.

However, the ASA might have localized network resources that are not defined globally that require local security groups with localized security policies. Local security groups can contain nested security groups that are downloaded from the ISE. The ASA consolidates local and central security groups.

To create local security groups on the ASA, you create a local security object group. A local security object group can contain one or more nested security object groups or Security IDs or security group names. You can also create a new Security ID or security group name that does not exist on the ASA.

You can use the security object groups you create on the ASA to control access to network resources. You can use the security object group as part of an access group or service policy.

Tip If you create a group with tags or names that are not known to the ASA, any rules that use the group will be inactive until the tags or names are resolved with ISE.

Procedure

Step 1 Create or edit a security group object group using the object name: **object-group security** *group_name*

Example:

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

Step 2 Add objects to the service group object group using one or more of the following commands. Use the **no** form of the command to remove an object.

- **security-group** {**tag** *sgt_number* | **name** *sg_name*}—A security group tag (SGT) or name. A tag is a number from 1 to 65533 and is assigned to a device through IEEE 802.1X authentication, web authentication, or MAC authentication bypass (MAB) by the ISE. Security group names are created on the ISE and provide user-friendly names for security groups. The security group table maps SGTs to security group names. Consult your ISE configuration for the valid tags and names.
- **group-object** *object_group_name*—The name of an existing security group object group.

Example:

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

Step 3 (Optional) Add a description: **description** *string*

Configure Time Ranges

A time range object defines a specific time consisting of a start time, an end time, and optional recurring entries. You use these objects on ACL rules to provide time-based access to certain features or assets. For example, you could create an access rule that allows access to a particular server during working hours only.

Note You can include multiple periodic entries in a time range object. If a time range has both absolute and periodic values specified, then the periodic values are evaluated only after the absolute start time is reached, and they are not further evaluated after the absolute end time is reached.

Creating a time range does not restrict access to the device. This procedure defines the time range only. You must then use the object in an access control rule.

Procedure

Step 1 Create the time range: **time-range** *name*

Step 2 (Optional.) Add a start or end time (or both) to the time range.

absolute [**start** *time date*] [**end** *time date*]

If you do not specify a start time, the default start time is now.

The *time* is in the 24-hour format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

The *date* is in the format *day month year*; for example, **1 January 2014**.

Step 3 (Optional.) Add recurring time periods.

periodic *days-of-the-week time* **to** [*days-of-the-week*] *time*

You can specify the following values for *days-of-the-week*. Note that you can specify a second day of the week only if you specify a single day for the first argument.

- **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, or **Sunday**. You can specify more than one of these, separated by spaces, for the first *days-of-the-week* argument.
- **daily**
- **weekdays**
- **weekend**

The *time* is in the 24-hour format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

You can repeat this command to configure more than one recurring period.

Examples

The following is an example of an absolute time range beginning at 8:00 a.m. on January 1, 2006. Because no end time and date are specified, the time range is in effect indefinitely.

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

The following is an example of a weekly periodic time range from 8:00 a.m. to 6:00 p.m on weekdays:

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

The following example establishes an end date for the time range, and sets a weekday period from 8 a.m. to 5 p.m., plus different hours after 5 for Monday, Wednesday, Friday compared to Tuesday, Thursday.

```
asa4(config)# time-range contract-A-access
asa4(config-time-range)# absolute end 12:00 1 September 2025
asa4(config-time-range)# periodic weekdays 08:00 to 17:00
asa4(config-time-range)# periodic Monday Wednesday Friday 18:00 to 20:00
asa4(config-time-range)# periodic Tuesday Thursday 17:30 to 18:30
```

Monitoring Objects

To monitor objects and groups, enter the following commands:

- **show access-list**

  Displays the access list entries. Entries that include objects are also expanded out into individual entries based on the object contents.

- **show running-config object** [**id** *object_id*]

  Displays all current objects. Use the **id** keyword to view a single object by name.

- **show running-config object** *object_type*

  Displays the current objects by their type, **network** or **service**.

- **show running-config object-group** [**id** *group_id*]

  Displays all current object groups. Use the **id** keyword to view a single object group by name.

- **show running-config object-group** *grp_type*

  Displays the current object groups by their group type.

History for Objects

Feature Name	Platform Releases	Description
Object groups	7.0(1)	Object groups simplify ACL creation and maintenance. We introduced or modified the following commands: **object-group** *protocol*, **object-group** *network*, **object-group** *service*, **object-group** *icmp_type*.
Regular expressions and policy maps	7.2(1)	Regular expressions and policy maps were introduced to be used under inspection policy maps. The following commands were introduced: **class-map type regex**, **regex**, **match regex**.
Objects	8.3(1)	Object support was introduced. We introduced or modified the following commands: **object-network**, **object-service**, **object-group** network, **object-group** service, **network object**, **access-list extended**, **access-list webtype**, **access-list remark**.
User Object Groups for Identity Firewall	8.4(2)	User object groups for identity firewall were introduced. We introduced the following commands: **object-network user**, **user**.
Security Group Object Groups for Cisco TrustSec	8.4(2)	Security group object groups for Cisco TrustSec were introduced. We introduced the following commands: **object-network security**, **security**.
Mixed IPv4 and IPv6 network object groups	9.0(1)	Previously, network object groups could only contain all IPv4 addresses or all IPv6 addresses. Now network object groups can support a mix of both IPv4 and IPv6 addresses. **Note** You cannot use a mixed object group for NAT. We modified the following commands: **object-group network**.
Extended ACL and object enhancement to filter ICMP traffic by ICMP code	9.0(1)	ICMP traffic can now be permitted/denied based on ICMP code. We introduced or modified the following commands: **access-list extended**, **service-object**, **service**.
Service object support for Stream Control Transmission Protocol (SCTP)	9.5(2)	You can now create service objects and groups that specific SCTP ports. We modified the following commands: **service-object**, **service**.