Basic Settings

This chapter describes how to configure basic settings on the ASA that are typically required for a functioning configuration.

Set the Hostname, Domain Name, and the Enable and Telnet Passwords

To set the hostname, domain name, and the enable and Telnet passwords, perform the following steps.

Before you begin

Before you set the hostname, domain name, and the enable and Telnet passwords, check the following requirements:

- In multiple context mode, you can configure the hostname and domain name in both the system and context execution spaces.
- For the enable and Telnet passwords, set them in each context; they are not available in the system.
- To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Specify the hostname for the ASA or for a context. The default hostname is “asa.”

hostname *name*

Example:

```
ciscoasa(config)# hostname myhostnamexample12345
```

This name can be up to 63 characters. The hostname must start and end with a letter or digit, and have only letters, digits, or a hyphen.

When you set a hostname for the ASA, that name appears in the command line prompt. If you establish sessions to multiple devices, the hostname helps you keep track of where you enter commands.

For multiple context mode, the hostname that you set in the system execution space appears in the command line prompt for all contexts. The hostname that you optionally set within a context does not appear in the command line, but can be used by the **banner** command **$(hostname)** token.

Step 2 Specify the domain name for the ASA. The default domain name is default.domain.invalid.

domain-name *name*

Example:

```
ciscoasa(config)# domain-name example.com
```

The ASA appends the domain name as a suffix to unqualified names. For example, if you set the domain name to "example.com" and specify a syslog server by the unqualified name of "jupiter," then the ASA qualifies the name to "jupiter.example.com."

Step 3 Change the enable password. By default, the enable password is blank, but you are prompted to change it the first time you enter the **enable** command.

enable password *password*

Example:

```
ciscoasa(config)# enable password Pa$$w0rd
```

The enable password lets you enter privileged EXEC mode if you do not configure enable authentication. The enable password also lets you log into ASDM with a blank username if you do not configure HTTP authentication.

The *password* argument is a case-sensitive password of 3 to 127 characters long, and can be any combination of ASCII printable characters (character codes 32-126), with the following exceptions:

- No spaces
- No question marks

This command changes the password for the highest privilege level (15). If you configure local command authorization, you can set enable passwords for each privilege level from 0 to 15 using the following syntax:

enable password *password* **level** *number*

The **encrypted** keyword (for passwords 32 characters and fewer in 9.6 and earlier) or the **pbkdf2** keyword (for passwords longer than 32 characters in 9.6 and later, and passwords of all lengths in 9.7 and later) indicates that the password is encrypted (using an MD5-based hash or a PBKDF2 (Password-Based Key Derivation Function 2) hash using SHA-512). Note that already existing passwords continue to use the MD5-based hash unless you enter a new password. When you define a password in the **enable password** command, the ASA

encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **enable password** command does not show the actual password; it shows the encrypted password followed by the **encrypted** or **pbkdf2** keyword. For example, if you enter the password "test," the **show running-config** command output would appear as something similar to the following:

```
username user1 password DLaUiAX3l78qgoB5c7iVNw== encrypted
```

The only time you would actually enter the **encrypted** or **pbkdf2** keyword at the CLI is if you are cutting and pasting a configuration file for use in another ASA, and you are using the same password.

You cannot reset the password to a blank value.

Step 4 Set the login password for Telnet access. There is no default password.

The login password is used for Telnet access when you do not configure Telnet authentication.

passwd *password* [**encrypted**]

Example:

```
ciscoasa(config)# passwd cisco12345
```

The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. If for some reason you need to copy the password to another ASA but do not know the original password, you can enter the **passwd** command with the encrypted password and the **encrypted** keyword. Normally, you only see this keyword when you enter the **show running-config passwd** command.

Set the Date and Time

Note Do not set the date and time for the Firepower 2100 in Platform mode, 4100/9300; the ASA receives these settings from the chassis.

Set the Time Zone and Daylight Saving Dates

To set the time zone and daylight saving date range, perform the following steps.

Procedure

Step 1 Set the time zone. By default, the time zone is UTC.

- Firepower models:

  clock timezone *zone*

- *zone*—Enter the **clock timezone ?** command to see a list of acceptable time zone names.

Example:

```
ciscoasa(config)# clock timezone ?
Available timezones:
CET
CST6CDT
Cuba
EET
Egypt
Eire
EST
EST5EDT
Factory
GB
GB-Eire
GMT
GMT0
GMT-0
GMT+0
Greenwich
Hongkong
HST
Iceland
Iran
Israel
Jamaica
Japan
[...]

ciscoasa(config)# clock timezone US/?

configure mode commands/options:
  US/Alaska          US/Aleutian    US/Arizona    US/Central
  US/East-Indiana    US/Eastern     US/Hawaii     US/Indiana-Starke
  US/Michigan        US/Mountain    US/Pacific

ciscoasa(config)# clock timezone US/Mountain
```

- All other models:

  clock timezone *zone* [-]*hours* [*minutes*]

  - *zone*—Specifies the time zone as a string, for example, PST for Pacific Standard Time.
  - [-]*hours*—Sets the number of hours of offset from UTC. For example, PST is -8 hours.
  - *minutes*—Sets the number of minutes of offset from UTC.

  Example:

```
ciscoasa(config)# clock timezone PST -8
```

Step 2 (ASA hardware, ASAv, and ISA 3000) Enter one of the following commands to change the date range for daylight saving time from the default. The default recurring date range is from 2:00 a.m. on the second Sunday in March to 2:00 a.m. on the first Sunday in November.

- Set the start and end dates for daylight saving time as a specific date in a specific year. If you use this command, you need to reset the dates every year.

  clock summer-time *zone* **date** {*day month* | *month day*} *year hh***:***mm* {*day month* | *month day*} *year hh***:***mm* [*offset*]

  - *zone* —Specifies the time zone as a string, for example, PDT for Pacific Daylight Time.
  - *day* —Sets the day of the month, from 1 to 31. You can enter the day and month as **A**pril 1 or as 1 April, for example, depending on your standard date format.
  - *month* —Sets the month as a string. You can enter the day and month as April 1 or as 1 April, depending on your standard date format.
  - *year* —Sets the year using four digits, for example, 2004. The year range is 1993 to 2035.
  - *hh:mm* —Sets the hour and minutes in 24-hour time.
  - *offset* —Sets the number of minutes to change the time for daylight saving time. By default, the value is 60 minutes.

  Example:

  ```
  ciscoasa(config)# clock summer-time PDT 1 April 2010 2:00 60
  ```

- Specify the start and end dates for daylight saving time, in the form of a day and time of the month, and not a specific date in a year. This command enables you to set a recurring date range that you do not need to change yearly.

  clock summer-time *zone* **recurring** [*week weekday month hh***:***mm week weekday month hh***:***mm*] [*offset*]

  - *zone*—Specifies the time zone as a string, for example, PDT for Pacific Daylight Time.
  - *week* —Specifies the week of the month as an integer between 1 and 4 or as the words first or last. For example, if the day might fall in the partial fifth week, then specify last.
  - *weekday* —Specifies the day of the week: Monday, Tuesday, Wednesday, and so on.
  - *month* —Sets the month as a string.
  - *hh:mm* —Sets the hour and minutes in 24-hour time.
  - *offset* —Sets the number of minutes to change the time for daylight savings time. By default, the value is 60 minutes.

  Example:

  ```
  ciscoasa(config)# clock summer-time PDT recurring first Monday April 2:00 60
  ```

Set the Date and Time Using an NTP Server

NTP is used to implement a hierarchical system of servers that provide a precisely synchronized time among network systems. This kind of accuracy is required for time-sensitive operations, such as validating CRLs, which include a precise time stamp. You can configure multiple NTP servers. The ASA chooses the server with the lowest stratum—a measure of how reliable the data is.

Time derived from an NTP server overrides any time set manually.

The ASA supports NTPv4.

Before you begin

In multiple context mode, you can set the time in the system configuration only.

Procedure

Step 1 (Optional) Enable authentication with an NTP server.

a) Enable authentication.

ntp authenticate

Example:

```
ciscoasa(config)# ntp authenticate
```

When you enable NTP authentication, you must also specify a key ID in the **ntp trusted-key** command and associate that key with the server with the **ntp server key** command. Configure the actual key for the ID with the **ntp authentication-key** command. If you have multiple servers, configure a separate ID for each sever.

b) Specify an authentication key ID to be a trusted key, which is required for authentication with an NTP server.

ntp trusted-key *key_id*

Example:

```
ciscoasa(config)# ntp trusted-key 1
ciscoasa(config)# ntp trusted-key 2
ciscoasa(config)# ntp trusted-key 3
ciscoasa(config)# ntp trusted-key 4
```

The *key_id* argument is a value between 1 and 4294967295. You can enter multiple trusted keys for use with multiple servers.

c) Set a key to authenticate with an NTP server.

ntp authentication-key *key_id* {**md5** | **sha1** | **sha256** | **sha512** | **cmac**} *key*

Example:

```
ciscoasa(config)# ntp authentication-key 1 md5 aNiceKey1
ciscoasa(config)# ntp authentication-key 2 md5 aNiceKey2
ciscoasa(config)# ntp authentication-key 3 md5 aNiceKey3
```

```
ciscoasa(config)# ntp authentication-key 4 md5 aNiceKey4
```

- *key_id*—Sets the ID that you set using the **ntp trusted-key** command.
- {**md5** | **sha1** | **sha256** | **sha512** | **cmac**} —Sets the algorithm.
- *key*—Sets the key as a string up to 32 characters long.

Step 2 Identify an NTP server.

ntp server {*ipv4_address* | *ipv6_address* }[**key** *key_id*] [**source** *interface_name*] [**prefer**]

Example:

```
ciscoasa(config)# ntp server 10.1.1.1 key 1 prefer
ciscoasa(config)# ntp server 10.2.1.1 key 2
ciscoasa(config)# ntp server 2001:DB8::178 key 3
ciscoasa(config)# ntp server 2001:DB8::8945:ABCD key 4
```

If you enabled NTP authentication (**ntp authenticate**), you must specify the **key** *key_id* argument using the ID that you set using the **ntp trusted-key** command.

The **source** *interface_name* keyword-argument pair identifies the outgoing interface for NTP packets if you do not want to use the default interface in the routing table. Because the system does not include any interfaces in multiple context mode, specify an interface name defined in the admin context.

The **prefer** keyword sets this NTP server as the preferred server if multiple servers have similar accuracy. NTP uses an algorithm to determine which server is the most accurate and synchronizes to that one. If servers are of similar accuracy, then the **prefer** keyword specifies which of those servers to use. However, if a server is significantly more accurate than the preferred one, the ASA uses the more accurate one. For example, the ASA uses a server of stratum 2 over a server of stratum 3 that is preferred.

You can identify multiple servers; the ASA uses the most accurate server.

Set the Date and Time Manually

To set the date and time manually, perform the following steps:

Before you begin

In multiple context mode, you can set the time in the system configuration only.

Procedure

Set the date time manually.

clock set *hh*:*mm*:*ss* {*month day* | *day month*} *year*

Example:

```
ciscoasa# clock set 20:54:00 april 1 2004
```

The *hh*:*mm*:*ss* argument sets the hour, minutes, and seconds in 24-hour time. For example, enter 20:54:00 for 8:54 pm.

The day value sets the day of the month, from 1 to 31. You can enter the day and month as april 1 or as 1 april, for example, depending on your standard date format.

The month value sets the month. Depending on your standard date format, you can enter the day and month as april 1 or as 1 april.

The year value sets the year using four digits, for example, 2004. The year range is from 1993 to 2035.

The default time zone is UTC. If you change the time zone after you enter the **clock set** command using the **clock timezone** command, the time automatically adjusts to the new time zone.

This command sets the time in the hardware chip, and does not save the time in the configuration file. This time endures reboots. Unlike the other **clock** commands, this command is a privileged EXEC command. To reset the clock, you need to set a new time with the clock set command.

Configure Precision Time Protocol (ISA 3000)

The Precision Time Protocol (PTP) is a time-synchronization protocol developed to synchronize the clocks of various devices in a packet-based network. These device clocks are generally of varying precision and stability. The protocol is designed specifically for industrial, networked measurement and control systems, and is optimal for use in distributed systems because it requires minimal bandwidth and little processing overhead.

A PTP system is a distributed, networked system consisting of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks and transparent clocks. Non-PTP devices include network switches, routers and other infrastructure devices.

You can configure the ASA device to be a transparent clock. The ASA device does not synchronize its clock with the PTP clocks. The ASA device will use the PTP default profile, as defined on the PTP clocks.

When you configure the PTP devices, you define a domain number for the devices that are meant to function together.Thus, you can configure multiple PTP domains, and then configure each non-PTP device to use the PTP clocks for one specific domain.

Note We added the following commands to the ASA default configuration to ensure that PTP traffic is not sent to the ASA FirePOWER module for inspection. If you have an existing deployment, you need to manually add these commands:

```
object-group service bypass_sfr_inspect
service-object udp destination range 319 320
access-list sfrAccessList extended deny object-group bypass_sfr_inspect any any
```

Before you begin

- This feature is only available on the ISA 3000.

- Use of PTP is supported in single context mode only.
- Cisco PTP supports multicast PTP messages only.
- PTP is enabled on all ISA 3000 interfaces in transparent mode by default. In routed mode, you must add the necessary configuration to ensure that the PTP packets are allowed to flow through the device.
- PTP is available only for IPv4 networks, not for IPv6 networks.
- PTP configuration is supported on physical Ethernet interfaces, whether stand-alone or bridge group members. It is not supported on:
  - Management interface.
  - Subinterfaces, EtherChannels, BVIs. or any other virtual interfaces.
- PTP flows on VLAN subinterfaces are supported, assuming the appropriate PTP configuration is present on the parent interface.
- You must ensure that PTP packets are allowed to flow through the device. In transparent firewall mode, the access list configuration to allow PTP traffic is configured by default. PTP traffic is identified by UDP ports 319 and 320, and destination IP address 224.0.1.129, so in routed firewall mode any ACL that allows this traffic should be acceptable.
- In routed firewall mode, you must also enable multicast routing for PTP multicast groups:
  - Enter the global configuration mode command **multicast-routing**.
  - And for each interface that is not a bridge group member, and on which PTP is enabled, enter the interface configuration command **igmp join-group 224.0.1.129** to statically enable PTP multicast group membership. This command is not supported or needed for bridge group members.

Procedure

Step 1 Specify the domain number of all ports of the device:

ptp domain *domain_num*

Example:

```
ciscoasa(config)# ptp domain 54
```

The *domain_num* argument is the domain number for all ports on the device. Packets received on a different domain are treated like regular multicast packets and will not undergo any PTP processing. This value can be from zero to 255; the default value is zero. Enter the domain number that is configured on the PTP devices in your network.

Step 2 (Optional) Configure the PTP clock mode on the device:

ptp mode e2etransparent

Example:

```
ciscoasa(config)# ptp mode e2etransparent
```

This command enables End-to-End Transparent mode on all PTP-enabled interfaces.

Step 3 Enable PTP on an interface:

ptp enable

Enable PTP on each interface through which the system can contact a PTP clock in the configured domain.

Example:

```
ciscoasa(config)# interface gigabitethernet1/2
ciscoasa(config-if)# ptp enable
```

Configure the Master Passphrase

The master passphrase allows you to securely store plain text passwords in encrypted format and provides a key that is used to universally encrypt or mask all passwords, without changing any functionality. Features that use the master passphrase include the following:

- OSPF
- EIGRP
- VPN load balancing
- VPN (remote access and site-to-site)
- Failover
- AAA servers
- Logging
- Shared licenses

Add or Change the Master Passphrase

To add or change the master passphrase, perform the following steps.

Before you begin

- This procedure will only be accepted in a secure session, for example by console, SSH, or ASDM via HTTPS.
- If failover is enabled but no failover shared key is set, an error message appears if you change the master passphrase, informing you that you must enter a failover shared key to protect the master passphrase changes from being sent as plain text.
- Enabling or changing password encryption in Active/Standby failover causes a **write standby**, which replicates the active configuration to the standby unit. Without this replication, the encrypted passwords on the standby unit will differ even though they use the same passphrase; configuration replication ensures that the configurations are the same. For Active/Active failover, you must manually enter **write standby**. A **write standby** can cause traffic interruption in Active/Active mode, because the configuration is cleared on the secondary unit before the new configuration is synced. You should make all contexts active on the primary ASA using the **failover active group 1** and **failover active group 2** commands,

enter **write standby**, and then restore the group 2 contexts to the secondary unit using the **no failover active group 2** command.

Procedure

Step 1 Set the passphrase used for generating the encryption key. The passphrase must be between 8 and 128 characters long. All characters except a backspace and double quotes are accepted for the passphrase. If you do not enter the new passphrase in the command, you are prompted for it. To change the passphrase, you must enter the old passphrase.

key config-key password-encryption [*new_passphrase* [*old_passphrase*]]

Example:

```
ciscoasa(config)# key config-key password-encryption
Old key: bumblebee
New key: haverford
Confirm key: haverford
```

Note
Use the interactive prompts to enter passwords to avoid having the passwords logged in the command history buffer.

Use the **no key config-key password-encrypt** command with caution, because it changes the encrypted passwords into plain text passwords. You may use the **no** form of this command when downgrading to a software version that does not support password encryption.

Step 2 Enable password encryption.

password encryption aes

Example:

```
ciscoasa(config)# password encryption aes
```

As soon as password encryption is enabled and the master passphrase is available, all the user passwords will be encrypted. The running configuration will show the passwords in the encrypted format.

If the passphrase is not configured at the time that password encryption is enabled, the command will succeed in anticipation that the passphrase will be available in the future.

If you later disable password encryption using the **no password encryption aes** command, all existing encrypted passwords are left unchanged, and as long as the master passphrase exists, the encrypted passwords will be decrypted, as required by the application.

Step 3 Save the runtime value of the master passphrase and the resulting configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

If you do not enter this command, passwords in startup configuration may still be visible if they were not saved with encryption previously. In addition, in multiple context mode the master passphrase is changed in the system context configuration. As a result, the passwords in all contexts will be affected. If the write memory command is not entered in the system context mode, but not in all user contexts, then the encrypted passwords in user contexts may be stale. Alternatively, use the write memory all command in the system context to save all configurations.

Examples

The following example shows that no previous key was present:

```
ciscoasa(config)# key config-key password-encryption 12345678
```

The following example shows that a key already exists:

```
ciscoasa(config)# key config-key password-encryption 23456789
Old key: 12345678
```

In the following example, you enter the command without parameters so that you will be prompted for keys. Because a key already exists, you are prompted for it.

```
ciscoasa(config)# key config-key password-encryption
Old key: 12345678
New key: 23456789
Confirm key: 23456789
```

In the following example, there is no existing key, so you are not prompted to supply it.

```
ciscoasa(config)# key config-key password-encryption
New key: 12345678
Confirm key: 12345678
```

Disable the Master Passphrase

Disabling the master passphrase reverts encrypted passwords into plain text passwords. Removing the passphrase might be useful if you downgrade to a previous software version that does not support encrypted passwords.

Before you begin

- You must know the current master passphrase to disable it. See Remove the Master Passphrase, on page 13 if you do not know the passphrase.
- This procedure works only in a secure session; that is, by Telnet, SSH, or ASDM via HTTPS.

To disable the master passphrase, perform the following steps:

Procedure

Step 1 Remove the master passphrase. If you do not enter the passphrase in the command, you are prompted for it.

no key config-key password-encryption [*old_passphrase*]]

Example:

```
ciscoasa(config)# no key config-key password-encryption

Warning! You have chosen to revert the encrypted passwords to plain text.
This operation will expose passwords in the configuration and therefore
exercise caution while viewing, storing, and copying configuration.

Old key: bumblebee
```

Step 2 Save the runtime value of the master passphrase and the resulting configuration.

write memory

Example:

```
ciscoasa(config)# write memory
```

The non-volatile memory containing the passphrase will be erased and overwritten with the 0xFF pattern.

In multiple mode, the master passphrase is changed in the system context configuration. As a result, the passwords in all contexts will be affected. If the write memory command is entered in the system context mode, but not in all user contexts, then the encrypted passwords in user contexts may be stale. Alternatively, use the write memory all command in the system context to save all configurations.

Remove the Master Passphrase

You cannot recover the master passphrase. If the master passphrase is lost or unknown, you can remove it.

To remove the master passphrase, perform the following steps:

Procedure

Step 1 Remove the master key and the configuration that includes the encrypted passwords.

write erase

Example:

```
ciscoasa(config)# write erase
```

Step 2 Reload the ASA with the startup configuration, without any master key or encrypted passwords.

reload

Example:

```
ciscoasa(config)# reload
```

Configure the DNS Server

You need to configure a DNS server so that the ASA can resolve host names to IP addresses. You also must configure a DNS server to use fully qualified domain names (FQDN) network objects in access rules.

Some ASA features require use of a DNS server to access external servers by domain name. Other features, such as the **ping** or **traceroute** command, let you enter a name that you want to ping or traceroute, and the ASA can resolve the name by communicating with a DNS server. Many SSL VPN and certificate commands also support names.

By default, there is a default DNS server group called DefaultDNS. You can create multiple DNS server groups: only one group can be active at a time, however.. Other DNS server groups can be configured for VPN tunnel groups. See the **tunnel-group** command in the command reference for more information.

Note The ASA has limited support for using the DNS server, depending on the feature. For example, most commands require you to enter an IP address and can only use a name when you manually configure the **name** command to associate a name with an IP address and enable use of the names using the names command.

Before you begin

Make sure that you configure the appropriate routing and access rules for any interface on which you enable DNS domain lookup so you can reach the DNS server.

Procedure

Step 1 Enable the ASA to send DNS requests to a DNS server to perform a name lookup for supported commands.

dns domain-lookup *interface_name*

If you do not enable DNS lookup on an interface, then the ASA will not communicate with the DNS server on that interface. Make sure to enable DNS lookup on all interfaces that will be used to access DNS servers.

Example:

```
ciscoasa(config)# dns domain-lookup inside
ciscoasa(config)# dns domain-lookup outside
```

Step 2 Create one or more DNS server groups and add servers to the groups.

a) Name the DNS server group.

dns server-group *name*

To configure the default DefaultDNS server group, specify DefaultDNS for the name.

Example:

```
ciscoasa(config)# dns server-group DefaultDNS
```

b) Specify one or more DNS servers for the group.

name-server *ip_address* [*ip_address2*] [...] [*ip_address6*] *[interface_name]*

You can enter all six IP addresses in the same command, separated by spaces, or you can enter each command separately.

(Optional) Specify the *interface_name* through which the ASA communicates with the server. If you do not specify the interface, the ASA checks the data routing table; if there are no matches, it then checks the management-only routing table.

The ASA tries each DNS server in order until it receives a response.

Example:

```
ciscoasa(config-dns-server-group)# name-server 10.1.1.5 192.168.1.67 209.165.201.6 outside
```

c) Configure the domain name appended to the hostname if it is not fully qualified.

domain-name *name*

Example:

```
ciscoasa(config-dns-server-group)# domain-name example.com
```

d) (Optional) Configure additional properties of the DNS server group.

Use the following commands to change the characteristics of the group, if the default settings are not appropriate for your network.

- **timeout** *seconds*—The number of seconds, from 1 to 30, to wait before trying the next DNS server. The default is 2 seconds. Each time the ASA retries the list of servers, this timeout doubles.
- **retries** *number*—The number of times, from 0 to 10, to retry the list of DNS servers when the ASA does not receive a response.
- **expire-entry-timer minutes** *number*—The number of minutes after a DNS entry expires (that is, the TTL has passed) that the entry is removed from the DNS lookup table. This command adds the specified value to the current TTL of the entry. Removing an entry requires that the table be recompiled, so frequent removals can increase the processing load on the device. Because some DNS entries can have very short TTL (as short as three seconds), you can use this setting to virtually extend the TTL. The default is 1 minute (that is, the entry is removed one minute after the TTL has passed). The range is 1 to 65535 minutes. This option is used when resolving FQDN network objects only.
- **poll-timer minutes** *number*—The time, in minutes, of the polling cycle used to resolve FQDN network/host objects to IP addresses. FQDN objects are resolved only if they are used in a firewall policy. The timer determines the maximum time between resolutions; the DNS entry's time-to-live (TTL) value is also used to determine when to update to IP address resolution, so individual FQDNs

might be resolved more frequently than the polling cycle. The default is 240 (four hours). The range is 1 to 65535 minutes.

e) Repeat the above steps to add additional DNS server groups.

Step 3 Specify the default DNS group.

dns-group *name*

By default, DefaultDNS is specified. If you configured other groups, you can specify a different default group using this command. You can only have one active group.

Example:

```
ciscoasa(config)# dns-group new_default_group
```

Configure the Hardware Bypass and Dual Power Supply (Cisco ISA 3000)

You can enable the hardware bypass so that traffic continues to flow between an interface pair during a power outage. Supported interface pairs are copper GigabitEthernet 1/1 & 1/2; and GigabitEthernet 1/3 & 1/4. When the hardware bypass is active, no firewall functions are in place, so make sure you understand the risks of allowing traffic through. See the following hardware bypass guidelines:

- This feature is only available on the Cisco ISA 3000 appliance.
- If you have a fiber Ethernet model, only the copper Ethernet pair (GigabitEthernet 1/1 & 1/2) supports hardware bypass.
- When the ISA 3000 loses power and goes into hardware bypass mode, only the supported interface pairs can communicate; when using the default configuration, inside1 <---> inside2, and outside1 <---> outside2 can no longer communicate. Any existing connections between these interfaces will be lost.
- We suggest that you disable TCP sequence randomization (as described in this procedure). If randomization is enabled (the default), then when the hardware bypass is activated, TCP sessions will need to be re-established. By default, the ISA 3000 rewrites the initial sequence number (ISN) of TCP connections passing through it to a random number. When the hardware bypass is activated, the ISA 3000 is no longer in the data path and does not translate the sequence numbers; the receiving client receives an unexpected sequence number and drops the connection. Even with TCP sequence randomization disabled, some TCP connections will have to be re-established because of the link that is temporarily down during the switchover.
- Cisco TrustSec connections on hardware bypass interfaces are dropped when hardware bypass is activated. When the ISA 3000 powers on and hardware bypass is deactivated, the connections are renegotiated.
- When the hardware bypass is deactivated, and traffic resumes going through the ISA 3000 data path, some existing TCP sessions need to be re-established because of the link that is temporarily down during the switchover.
- When hardware bypass is active, the Ethernet PHYs are disconnected, so the ASA is unable to determine the interface status. Interfaces may appear to be in a down state.

For dual power supplies in the ISA 3000, you can establish dual power supplies as the expected configuration in the ASA OS. If one power supply fails, the ASA issues an alarm. By default, the ASA expects a single power supply and won't issue an alarm as long as it includes one working power supply.

Before you begin

- You must attach the hardware bypass interfaces to access ports on the switch. Do not attach them to trunk ports.

Procedure

Step 1 Configure the hardware bypass to activate during a power failure:

hardware-bypass GigabitEthernet {**1/1-1/2** | **1/3-1/4**} [**sticky**]

Example:

```
ciscoasa(config)# hardware-bypass GigabitEthernet 1/1-1/2
ciscoasa(config)# hardware-bypass GigabitEthernet 1/3-1/4
```

The **sticky** keyword keeps the appliance in hardware bypass mode after the power comes back and the appliance boots up. In this case, you need to manually turn off the hardware bypass when you are ready; this option lets you control when the brief interruption in traffic occurs.

Step 2 Manually activate or deactivate the hardware bypass:

[**no**] **hardware-bypass manual GigabitEthernet** {**1/1-1/2** | **1/3-1/4**}

Example:

```
ciscoasa# hardware-bypass manual GigabitEthernet 1/1-1/2
ciscoasa# no hardware-bypass manual GigabitEthernet 1/1-1/2
```

Step 3 (Optional) Configure the hardware bypass to remain active until after the ASA FirePOWER module boots up:

hardware-bypass boot-delay module-up sfr

You must enable hardware bypass without the **sticky** option for the boot delay to operate. Without the **hardware-bypass boot-delay** command, the hardware bypass is likely to become inactive before the ASA FirePOWER module finishes booting up. This scenario can cause traffic to be dropped if you configured the module to fail-close, for example.

Step 4 Disable TCP sequence randomization. This example shows how to disable randomization for all traffic by adding the setting to the default configuration.

policy-map global_policy

class sfrclass

set connection random-sequence-number disable

If you later decide to turn it back on, replace "disable" with **enable**.

Step 5 Establish dual power supplies as the expected configuration:

power-supply dual

Step 6 Save the configuration.

write memory

The behavior of hardware bypass after the system comes online is determined by the configuration setting in the startup configuration, so you must save your running configuration.

Adjust ASP (Accelerated Security Path) Performance and Behavior

The ASP is an implementation layer that puts your policies and configurations into action. It is not of direct interest except during troubleshooting with the Cisco Technical Assistance Center. However, there are a few behaviors related to performance and reliability that you can adjust.

Choose a Rule Engine Transactional Commit Model

By default, when you change a rule-based policy (such as access rules), the changes become effective immediately. However, this immediacy comes with a slight cost in performance. The performance cost is more noticeable for very large rule lists in a high connections-per-second environment, for example, when you change a policy with 25,000 rules while the ASA is handling 18,000 connections per second.

The performance is affected because the rule engine compiles rules to enable faster rule lookup. By default, the system also searches uncompiled rules when evaluating a connection attempt so that new rules can be applied; because the rules are not compiled, the search takes longer.

You can change this behavior so that the rule engine uses a transactional model when implementing rule changes, continuing to use the old rules until the new rules are compiled and ready for use. With the transactional model, performance should not drop during the rule compilation. The following table clarifies the behavioral difference.

Model	Before Compilation	During Compilation	After Compilation
Default	Matches old rules.	Match new rules. (The rate for connections per second decreases.)	Matches new rules.
Transactional	Matches old rules.	Match old rules. (The rate for connections per second is unaffected.)	Matches new rules.

An additional benefit of the transactional model is that, when replacing an ACL on an interface, there is no gap between deleting the old ACL and applying the new one. This feature reduces the chances that acceptable connections may be dropped during the operation.

Before you begin

- Transactional commit is not recommended for access control rules if you use FQDN objects for hostnames whose resolution can frequently change, as the access group compilation might never completely resolve

due to DNS churn. If you still want to use transactional commit, consider lengthening the time for DNS expiration.

- If you enable the transactional model for a rule type, syslogs to mark the beginning and the end of the compilation are generated. These syslogs are numbered 780001 through 780004.

Procedure

Enable the transactional commit model for the rule engine:

asp rule-engine transactional-commit *option*

Where the options are:

- **access-group**—Access rules applied globally or to interfaces.
- **nat**—Network Address Translation rules.

Example:

```
ciscoasa(config)# asp rule-engine transactional-commit access-group
```

Enable ASP Load Balancing

The ASP load balancing mechanism helps avoid the following issues:

- Overruns caused by sporadic traffic spikes on flows
- Overruns caused by bulk flows oversubscribing specific interface receive rings
- Overruns caused by relatively heavily overloaded interface receive rings, in which a single core cannot sustain the load.

ASP load balancing allows multiple cores to work simultaneously on packets that were received from a single interface receive ring. If the system drops packets, and the **show cpu** command output is far less than 100%, then this feature may help your throughput if the packets belong to many unrelated connections.

Note ASP load balancing is disabled on the ASAv. With the integration of DPDK (Dataplane Development Kit) into the ASAv's accelerated security path (ASP), the ASAv shows better performance with this feature disabled.

Procedure

Step 1 Enable the automatic switching on and off of ASP load balancing:

asp load-balance per-packet auto

Step 2 Manually enable ASP load balancing:

asp load-balance per-packet

ASP load balancing is enabled until you manually disable it, even if you also have the **auto** command enabled.

Step 3 Manually disable ASP load balancing:

no asp load-balance per-packet

This command only applies if you manually enabled ASP load blancing. If you also enabled the **auto** command, then the system reverts to automatically enabling or disabling ASP load balancing.

Monitoring the DNS Cache

The ASA provides a local cache of DNS information from external DNS queries that are sent for certain clientless SSL VPN and certificate commands. Each DNS translation request is first looked for in the local cache. If the local cache has the information, the resulting IP address is returned. If the local cache can not resolve the request, a DNS query is sent to the various DNS servers that have been configured. If an external DNS server resolves the request, the resulting IP address is stored in the local cache with its corresponding hostname.

See the following command for monitoring the DNS cache:

- **show dns-hosts**

  This command shows the DNS cache, which includes dynamically learned entries from a DNS server as well as manually entered name and IP addresses using the name command.

History for Basic Settings

Feature Name	Platform Releases	Description
NTPv4 support	9.14(1)	The ASA now supports NTPv4. No modified commands.
Additional NTP authentication algorithms	9.13(1)	Formerly, only MD5 was supported for NTP authentication. The ASA now supports the following algorithms: • MD5 • SHA-1 • SHA-256 • SHA-512 • AES-CMAC New/Modified commands: **ntp authentication-key**

Feature Name	Platform Releases	Description
NTP support on IPv6	9.12(1)	You can now specify an IPv6 address for the NTP server. New/Modified commands: **ntp server**
enable password change now required on login	9.12(1)	The default **enable** password is blank. When you try to access privileged EXEC mode on the ASA, you are now required to change the password to a value of 3 to 127 characters. You cannot keep it blank. The **no enable password** command is no longer supported. At the CLI, you can access privileged EXEC mode using the **enable** command, the **login** command (with a user at privilege level 2+), or an SSH or Telnet session when you enable **aaa authorization exec auto-enable**. All of these methods require you to set the enable password. This password change requirement is not enforced for ASDM logins. In ASDM, by default you can log in without a username and with the **enable** password. New/Modified commands: **enable password**
ASP load balancing is disabled on the ASAv	9.10(1)	With the recent integration of DPDK (Dataplane Development Kit) into the ASAv's accelerated security path (ASP), the ASAv shows better performance with this feature disabled.
Automatic ASP load balancing now supported for the ASAv	9.8(1)	Formerly, you could only manually enable and disable ASP load balancing. We modified the following command: **asp load-balance per-packet auto**
PBKDF2 hashing for all local **username** and **enable** passwords	9.7(1)	Local **username** and **enable** passwords of all lengths are stored in the configuration using a PBKDF2 (Password-Based Key Derivation Function 2) hash using SHA-512. Previously, passwords 32 characters and shorter used the MD5-based hashing method. Already existing passwords continue to use the MD5-based hash unless you enter a new password. See the "Software and Configurations" chapter in the General Operations Configuration Guide for downgrading guidelines. We modified the following commands: **enable, username**
Dual power supply support for the ISA 3000	9.6(1)	For dual power supplies in the ISA 3000, you can establish dual power supplies as the expected configuration in the ASA OS. If one power supply fails, the ASA issues an alarm. By default, the ASA expects a single power supply and won't issue an alarm as long as it includes one working power supply. We introduced the following command: **power-supply dual**
Longer password support for local **username** and **enable** passwords (up to 127 characters)	9.6(1)	You can now create local **username** and **enable** passwords up to 127 characters (the former limit was 32). When you create a password longer than 32 characters, it is stored in the configuration using a PBKDF2 (Password-Based Key Derivation Function 2) hash. Shorter passwords continue to use the MD5-based hashing method. We modified the following commands: **enable, username**
ISA 3000 hardware bypass	9.4(1.225)	The ISA 3000 supports a hardware bypass function to allow traffic to continue flowing through the appliance when there is a loss of power. We introduced the following commands: **hardware-bypass, hardware-bypass manual, hardware-bypass boot-delay, show hardware-bypass** *This feature is not available in Version 9.5(1).*

Feature Name	Platform Releases	Description
Automatic ASP Load Balancing	9.3(2)	You can now enable automatic switching on and off of the ASP load balancing feature. **Note** The automatic feature is not supported on the ASAv; only manual enabling and disabling is supported. We introduced the following command: **asp load-balance per-packet auto**.
Removal of the default Telnet password	9.0(2), 9.1(2)	To improve security for management access to the ASA, the default login password for Telnet was removed; you must manually set the password before you can log in using Telnet. **Note** The login password is only used for Telnet if you do not configure Telnet user authentication (the **aaa authentication telnet console** command). Previously, when you cleared the password, the ASA restored the default of "cisco." Now when you clear the password, the password is removed. The login password is also used for Telnet sessions from the switch to the ASASM (see the **session** command). For initial ASASM access, you must use the **service-module session** command, until you set a login password. We modified the following command: **password**
Password Encryption Visibility	8.4(1)	We modified the **show password encryption** command.
Master Passphrase	8.3(1)	We introduced this feature. The master passphrase allows you to securely store plain text passwords in encrypted format and provides a key that is used to universally encrypt or mask all passwords, without changing any functionality. We introduced the following commands: **key config-key password-encryption**, **password encryption aes**, **clear configure password encryption aes**, **show running-config password encryption aes, show password encryption**