



CHAPTER 38

Managing Software, Licenses, and Configurations

This chapter contains information about managing the adaptive security appliance software, licenses, and configurations, and includes the following sections:

- [Managing Licenses, page 38-1](#)
- [Viewing Files in Flash Memory, page 38-3](#)
- [Retrieving Files from Flash Memory, page 38-3](#)
- [Downloading Software or Configuration Files to Flash Memory, page 38-4](#)
- [Performing Zero Downtime Upgrades for Failover Pairs, page 38-6](#)
- [Configuring the Application Image and ASDM Image to Boot, page 38-8](#)
- [Configuring a File to Boot as the Startup Configuration, page 38-8](#)
- [Backing Up Configuration Files, page 38-9](#)
- [Configuring Auto Update Support, page 38-20](#)

Managing Licenses

When you install the software, the existing activation key is extracted from the original image and stored in a file in the adaptive security appliance file system.

This section includes the following topics:

- [Obtaining an Activation Key, page 38-2](#)
- [Entering a New Activation Key, page 38-2](#)



Note

During initial start up on a Cisco ASA 5580 Series adaptive security appliance, the software checks for a valid license key. If none is found, the system sends the following prompt: “There is no valid license key. Please press ESC or SPACE to enter key. The system will reboot in 10 seconds.” If the proper license key is not entered at this point, the system will reboot instead of operating with a minimal feature set.

Obtaining an Activation Key

To obtain an activation key, you will need a Product Authorization Key, which you can purchase from your Cisco account representative. After obtaining the Product Authorization Key, register it on the Web by performing the following steps:

Step 1 Obtain the serial number for your adaptive security appliance by entering the following command:

```
hostname> show version | include Number
```

Enter the pipe character (|) as part of the command.

Step 2 Open a web browser and go to one of the following websites (the URLs are case-sensitive):

If you are a registered user of Cisco.com, go to the following URL:

```
http://www.cisco.com/go/license
```

If you are not a registered user of Cisco.com, go to the following URL:

```
http://www.cisco.com/go/license/public
```

Step 3 When prompted, enter the following information:

- Your Product Authorization Key
- The serial number of your adaptive security appliance
- Your e-mail address

The activation key will be automatically generated and sent to the e-mail address that you provide.

Entering a New Activation Key

To enter the activation key, enter the following command:

```
hostname(config)# activation-key key
```

where *key* is a four or five-element hexadecimal string with one space between each element. For example:

```
0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e
```

The leading 0x specifier is optional; all values are assumed to be hexadecimal.

If you are already in multiple context mode, enter this command in the system execution space.

Before entering the activation key, ensure that the image in flash memory and the running image are the same. You can do this by rebooting the adaptive security appliance before entering the new activation key.



Note

The activation key is not stored in your configuration file. The key is tied to the serial number of the security appliance.

You must reboot the adaptive security appliance after entering the new activation key for the change to take effect in the running image.

The following example shows how to change the activation key on the adaptive security appliance:

```
hostname(config)# activation-key 0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e
```

Viewing Files in Flash Memory

You can view files in flash memory and see information about the files.

- To view the files in flash memory, enter the following command:

```
hostname# dir [flash: | disk0: | disk1:]
```

You can enter **flash:** or **disk0:** for the internal flash memory on the ASA 5580. The **disk1:** keyword represents the external flash memory on the ASA 5580. The internal flash memory is the default.

For example:

```
hostname# dir

Directory of disk0:/
500  -rw-  4958208    22:56:20 Nov 29 2006  cdisk.bin
2513 -rw-   4634      19:32:48 Sep 17 2006  first-backup
2788 -rw-   21601    20:51:46 Nov 23 2006  backup.cfg
2927 -rw-  8670632    20:42:48 Dec 08 2006  asdmfile.bin
```

- To view extended information about a specific file, enter the following command:

```
hostname# show file information [path:/] filename
```

The default path is the root directory of the internal flash memory (flash:/ or disk0:/).

For example:

```
hostname# show file information cdisk.bin

disk0:/cdisk.bin:
  type is image (XXX) []
  file size is 4976640 bytes version 8.1(1)
```

The file size listed is for example only.

Retrieving Files from Flash Memory

You can retrieve files directly from the Flash disk by using an HTTPS connection with the following URL, in which you supply values for the ASA IP address and the filename:

```
https://ASA_IP/disk0/filename
```

This option is useful for customers who wish to do the following:

- Copy from the ASA binary image (as a backup).
- Copy from WebVPN capture files.
- Copy any other Flash files to a secure desktop.

Downloading Software or Configuration Files to Flash Memory

This section includes the following topics:

- [Downloading a File to a Specific Location, page 38-4](#)
- [Downloading a File to the Startup Configuration or Running Configuration, page 38-5](#)

You can download application images, ASDM images, configuration files, and other files to the internal flash memory or, for the ASA 5580, to the external flash memory from a TFTP, FTP, HTTP, or HTTPS server.



Note

You cannot have two files with the same name but with different letter case in the same directory in flash memory. For example, if you try to download the file, Config.cfg to a location that contains the file, config.cfg, you receive the following error: %Error opening disk0:/Config.cfg (File exists).

Downloading a File to a Specific Location

This section describes how to download the application image, ASDM software, a configuration file, or any other file that needs to be downloaded to flash memory. To download a file to the startup configuration or running configuration, see the “[Downloading a File to the Startup Configuration or Running Configuration](#)” section on page 38-5.

For information about installing the Cisco SSL VPN client, see the “[Installing the AnyConnect Client](#)” section on page 39-2. For information about installing Cisco Secure Desktop on the security appliance, see the *Cisco Secure Desktop Configuration Guide for Cisco ASA 5580 Administrators*.

To configure the adaptive security appliance to use a specific application image or ASDM image if you have more than one installed, or have installed them in external flash memory, see the “[Configuring the Application Image and ASDM Image to Boot](#)” section on page 38-8.



Note

To successfully copy ASDM Version 6.1(1) to flash memory, you must be running Version 8.1(1).

To configure the adaptive security appliance to use a specific configuration as the startup configuration, see the “[Configuring a File to Boot as the Startup Configuration](#)” section on page 38-8.

For multiple context mode, you must be in the system execution space.

To download a file to flash memory, choose one of the following commands for the appropriate download server type:

- To copy from a TFTP server, enter the following command:

```
hostname# copy tftp://server[/path]/filename {flash:/ | disk0:/ |
disk1:/}[path/]filename
```

You can enter **flash:/** or **disk0:/** for the internal flash memory on the ASA 5580. The **disk1:/** keyword represents the external flash memory on the security appliance.

- To copy from an FTP server, enter the following command:

```
hostname# copy ftp://[user[:password]@]server[/path]/filename {flash:/ | disk0:/ |
disk1:/} [path/] filename
```

- To copy from an HTTP or HTTPS server, enter the following command:

```
hostname# copy http[s]://[user[:password]@]server[:port]/[path]/filename {flash:/ |
disk0:/ | disk1:/} [path/] filename
```

- To use secure copy, first enable SSH, then enter the following command:

```
hostname# ssh scopy enable
```

Then from a Linux client, enter the following command:

```
scp -v -pw password filename username@asa_address
```

The **-v** is for verbose, and if **-pw** is not specified, you will be prompted for a password.

Downloading a File to the Startup Configuration or Running Configuration

You can download a text file to the startup configuration or running configuration from a TFTP, FTP, or HTTP(S) server, or from the flash memory.



Note

When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

To copy a file to the startup configuration or running configuration, choose one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

```
hostname# copy tftp://server[/path]/filename {startup-config | running-config}
```

- To copy from an FTP server, enter the following command:

```
hostname# copy ftp://[user[:password]@]server[/path]/filename {startup-config |
running-config}
```

- To copy from an HTTP or HTTPS server, enter the following command:

```
hostname# copy http[s]://[user[:password]@]server[:port]/[path]/filename
{startup-config | running-config}
```

- To copy from flash memory, enter the following command:

```
hostname# copy {flash:/ | disk0:/ | disk1:/} [path/] filename
{startup-config | running-config}
```

For example, to copy the configuration from a TFTP server, enter the following command:

```
hostname# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
hostname# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
hostname# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Performing Zero Downtime Upgrades for Failover Pairs

The two units in a failover configuration should have the same major (first number) and minor (second number) software version. However, you do not need to maintain version parity on the units during the upgrade process; you can have different versions on the software running on each unit and still maintain failover support. To ensure long-term compatibility and stability, we recommend upgrading both units to the same version as soon as possible.

[Table 38-1](#) shows the supported scenarios for performing zero-downtime upgrades on a failover pair.

Table 38-1 Zero-Downtime Upgrade Support

Type of Upgrade	Support
Maintenance Release	You can upgrade from any maintenance release to any other maintenance release within a minor release. For example, you can upgrade from 7.0(1) to 7.0(4) without first installing the maintenance releases in between.
Minor Release	You can upgrade from a minor release to the next minor release. You cannot skip a minor release. For example, you can upgrade from 7.0 to 7.1. Upgrading from 7.0 directly to 7.2 is not supported for zero-downtime upgrades; you must first upgrade to 7.1.
Major Release	You can upgrade from the last minor release of the previous version to the next major release. For example, you can upgrade from 7.9 to 8.0, assuming that 7.9 is the last minor version in the 7.x release.


For more details about upgrading the software on a failover pair, refer to the following topics:

- [Upgrading an Active/Standby Failover Configuration, page 38-6](#)
- [Upgrading and Active/Active Failover Configuration, page 38-7](#)

Upgrading an Active/Standby Failover Configuration


To upgrade two units in an Active/Standby failover configuration, perform the following steps:

-
- Step 1** Download the new software to both units, and specify the new image to load with the **boot system** command (see the [“Configuring the Application Image and ASDM Image to Boot”](#) section on [page 38-8](#)).

- Step 2** Reload the standby unit to boot the new image by entering the following command on the active unit:
- ```
active# failover reload-standby
```
- Step 3** When the standby unit has finished reloading, and is in the Standby Ready state, force the active unit to fail over to the standby unit by entering the following command on the active unit.
-  **Note** Use the **show failover** command to verify that the standby unit is in the Standby Ready state.
- ```
active# no failover active
```
- Step 4** Reload the former active unit (now the new standby unit) by entering the following command:
- ```
newstandby# reload
```
- Step 5** When the new standby unit has finished reloading, and is in the Standby Ready state, return the original active unit to active status by entering the following command:
- ```
newstandby# failover active
```

Upgrading and Active/Active Failover Configuration

To upgrade two units in an Active/Active failover configuration, perform the following steps:

- Step 1** Download the new software to both units, and specify the new image to load with the **boot system** command (see the [“Configuring the Application Image and ASDM Image to Boot”](#) section on [page 38-8](#)).
- Step 2** Make both failover groups active on the primary unit by entering the following command in the system execution space of the primary unit:
- ```
primary# failover active
```
- Step 3** Reload the secondary unit to boot the new image by entering the following command in the system execution space of the primary unit:
- ```
primary# failover reload-standby
```
- Step 4** When the secondary unit has finished reloading, and both failover groups are in the Standby Ready state on that unit, make both failover groups active on the secondary unit using the following command in the system execution space of the primary unit:
-  **Note** Use the **show failover** command to verify that both failover groups are in the Standby Ready state on the secondary unit.
- ```
primary# no failover active
```
- Step 5** Make sure both failover groups are in the Standby Ready state on the primary unit, and then reload the primary unit using the following command:
- ```
primary# reload
```

- Step 6** If the failover groups are configured with the **preempt** command, they will automatically become active on their designated unit after the preempt delay has passed. If the failover groups are not configured with the **preempt** command, you can return them to active status on their designated units using the **failover active group** command.

Configuring the Application Image and ASDM Image to Boot

By default, the adaptive security appliance boots the first application image it finds in internal flash memory. The security appliance also boots the first ASDM image it finds in internal flash memory, or if none exists, then in external flash memory. If you have more than one image, you should specify the image that you want to boot. For the ASDM image, if you do not specify the image to boot, even if you have only one image installed, then the adaptive security appliance inserts the **asdm image** command into the running configuration. To avoid problems with Auto Update (if configured), and to avoid the image search at each startup, you should specify the ASDM image that you want to boot in the startup configuration.

- To configure the application image to boot, enter the following command:

```
hostname(config)# boot system url
```

where *url* is one of the following:

- **{flash:/ | disk0:/ | disk1:/}***[path/]filename*

You can enter **flash:/** or **disk0:/** for the internal flash memory on the ASA 5580. The **disk1:/** keyword represents the external flash memory on the security appliance.

- **tftp://***[user[:password]@]server[:port]/[path/]filename*

This option is only supported for the ASA 5580.

You can enter up to four **boot system** commands to specify different images to boot from in order; the adaptive security appliance boots the first image it finds. You can configure only one **boot system tftp:** command, and it must be the first one configured.

- To configure the ASDM image to boot, enter the following command:

```
hostname(config)# asdm image {flash:/ | disk0:/ | disk1:/}[path/]filename
```

Configuring a File to Boot as the Startup Configuration

By default, the adaptive security appliance boots from a startup configuration that is a hidden file. You can alternatively set any configuration to be the startup configuration by entering the following command:

```
hostname(config)# boot config {flash:/ | disk0:/ | disk1:/}[path/]filename
```

You can enter **flash:/** or **disk0:/** for the internal flash memory on the ASA 5580. The **disk1:/** keyword represents the external flash memory on the security appliance.

Backing Up Configuration Files

To back up your configuration, use one of the following methods:

- [Backing up a Single Mode Configuration or Multiple Mode System Configuration, page 38-9](#)
- [Backing Up a Context Configuration in Flash Memory, page 38-9](#)
- [Backing Up a Context Configuration within a Context, page 38-10](#)
- [Copying the Configuration from the Terminal Display, page 38-10](#)
- [Backing Up Additional Files Using the Export and Import Commands, page 38-10](#)
- [Using a Script to Back Up and Restore Files, page 38-11](#)

Backing up a Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local flash memory:

- To copy to a TFTP server, enter the following command:

```
hostname# copy {startup-config | running-config} tftp://server[/path]/filename
```

- To copy to a FTP server, enter the following command:

```
hostname# copy {startup-config | running-config}
ftp://[user[:password]@]server[/path]/filename
```

- To copy to local flash memory, enter the following command:

```
hostname# copy {startup-config | running-config} {flash:/ | disk0:/ |
disk1:/} [path/] filename
```

Be sure the destination directory exists. If it does not exist, create the directory using the **mkdir** command.

Backing Up a Context Configuration in Flash Memory

In multiple context mode, copy context configurations that are on the local flash memory by entering one of the following commands in the system execution space:

- To copy to a TFTP server, enter the following command:

```
hostname# copy disk:[path/] filename tftp://server[/path]/filename
```

- To copy to a FTP server, enter the following command:

```
hostname# copy disk:[path/] filename ftp://[user[:password]@]server[/path]/filename
```

- To copy to local flash memory, enter the following command:

```
hostname# copy {flash:/ | disk0:/ | disk1:/} [path/] filename {flash:/ | disk0:/ |
disk1:/} [path/] newfilename
```

Be sure the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

Backing Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups:

- To copy the running configuration to the startup configuration server (connected to the admin context), enter the following command:

```
hostname/contexta# copy running-config startup-config
```

- To copy the running configuration to a TFTP server connected to the context network, enter the following command:

```
hostname/contexta# copy running-config tftp://server[/path]/filename
```

Copying the Configuration from the Terminal Display

To print the configuration to the terminal, enter the following command:

```
hostname# show running-config
```

Copy the output from this command, then paste the configuration into a text file.

Backing Up Additional Files Using the Export and Import Commands

Additional files essential to your configuration might include the following:

- Files you import using the **import webvpn** command. Currently these files include customizations, URL lists, web contents, plug-ins, and language translations.
- mDAP policies (dap.xml)
- CSD configurations (data.xml)
- Digital keys and certificates
- Local CA user database and certificate status files

The CLI lets you back up and restore individual elements of your configuration using the **export** and **import** commands. To back up these files, for example, those imported via the **import webvpn** command or certificates, follow these steps:

Step 1 Issue the appropriate **show** command(s). For example:

```
hostname# show import webvpn plug-in
ica
rdp
ssh, telnet
vnc
hostname#
```

Step 2 Issue the **export** command for the file you want to back up:

```
hostname# export webvpn plug-in protocol rdp tftp://tftpserver/backupfilename
hostname#
```

Using a Script to Back Up and Restore Files

This section includes the following topics:

- [Prerequisites, page 38-11](#)
- [Running the Script, page 38-11](#)

You can use a script to back up and restore the configuration files on your security appliance, including all of the extensions you import via the **import webvpn** CLI, the CSD configuration XML files, and the DAP configuration XML file. For security reasons, we do not recommend that you perform automated backups of digital keys and certificates or the Local CA key.

This section includes a sample script that you can use as-is or change as your environment requires. The sample script is specific to a Linux system. To use it for a Microsoft Windows system, you need to modify it using the logic of the sample.



Note

The existing CLI lets you back up and restore individual files using the **copy**, **export**, and **import** commands. However, there is no facility that lets you back up all security appliance configuration files in one operation. Running the script facilitates the use of multiple CLIs.

Prerequisites

To use a script to back up and restore a security appliance configuration, first perform the following tasks:

- Install Perl with an Expect module.
- Install an SSH client that can reach the security appliance.
- Install a TFTP server to send files from the security appliance to the backup location.

Another option is to use a commercially available tool. You can include this script in such a tool.

Running the Script

To run a backup and restore script, follow these steps:

-
- Step 1** Download or cut and paste the script file to any location on your system.
 - Step 2** At the command line, enter **Perl *scriptname***, where *scriptname* is the name of the script file.
 - Step 3** Press **Enter**.
 - Step 4** The system prompts you for values for each of the options. Alternatively, you can enter values for the options when you enter the **Perl *scriptname*** command. You must enter a value for each option.
- The script starts running, printing out the commands that it issues, which provides a record of the CLIs. You can use these CLIs for a later restore operation.
-

Sample Script

```
#!/usr/bin/perl
#Function: Backup/restore configuration/extensions to/from a TFTP server.
```

```

#Description: The objective of this script is to show how to back up configurations/extensions
# before the backup/restore command is developed.
# It currently backs up the running configuration, all extensions imported via the "import webvpn"
# command, the CSD configuration XML file, and the DAP configuration XML file.
#Requirements: Perl with Expect, SSH to the ASA, and a TFTP server.
#Usage: backupasa -option option_value
# -h: ASA hostname or IP address
# -u: User name to log in via SSH
# -w: Password to log in via SSH
# -e: The Enable password on the security appliance
# -p: Global configuration mode prompt
# -s: Host name or IP address of the TFTP server to store the configurations
# -r: Restore with an argument that specifies the file name. This file is produced during backup.
#If you do not enter an option, the script will prompt for it before backup.
#
#Make sure that you can SSH to the ASA.

use Expect;
use Getopt::Std;

#global variables
%options=();
$restore = 0; #does backup by default
$restore_file = "";
$sasa = "";
$storage = "";
$user = "";
$password = "";
$enable = "";
$prompt = "";
$date = `date +%F`;
chop($date);
my $exp = new Expect();

getopts("h:u:p:w:e:s:r:", \%options);
do process_options();

do login($exp);

```

```
do enable($exp);
if ($restore) {
    do restore($exp,$restore_file);
}
else {
    $restore_file = "$prompt-restore-$date.cli";
    open(OUT,">$restore_file") or die "Can't open $restore_file\n";
    do running_config($exp);
    do lang_trans($exp);
    do customization($exp);
    do plugin($exp);
    do url_list($exp);
    do webcontent($exp);
    do dap($exp);
    do csd($exp);
    close(OUT);
}
do finish($exp);

sub enable {
    $obj = shift;
    $obj->send("enable\n");
    unless ($obj->expect(15, 'Password:')) {
        print "timed out waiting for Password:\n";
    }
    $obj->send("$enable\n");
    unless ($obj->expect(15, "$prompt#")) {
        print "timed out waiting for $prompt#\n";
    }
}

sub lang_trans {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn translation-table\n");
    $obj->expect(15, "$prompt#");
    $output = $obj->before();
    @items = split(/\n+/, $output);
```

```

for (@items) {
    s/^s+//;
    s/s+$//;
    next if /show import/ or /Translation Tables/;
    next unless (/^.\s+.\s+$/);
    ($lang, $transtable) = split(/\s+/, $_);
    $cli = "export webvpn translation-table $transtable language $lang
$storage/$prompt-$date-$transtable-$lang.po";
    $ocli = $cli;
    $ocli =~ s/^export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}
}

```

```

sub running_config {
    $obj = shift;
    $obj->clear_accum();
    $cli = "copy /noconfirm running-config $storage/$prompt-$date.cfg";
    print "$cli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

```

```

sub customization {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn customization\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
    }
}

```

```
next if /^Template/ or /show import/ or /\s*$/;
$ccli = "export webvpn customization $_ $storage/$prompt-$date-cust-$_.xml";
$occli = $ccli;
$occli =~ s/^export/import/;
print "$ccli\n";
print OUT "$occli\n";
$obj->send("$ccli\n");
$obj->expect(15, "$prompt#" );
}
}

sub plugin {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn plug-in\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        chop;
        next if /^Template/ or /show import/ or /\s*$/;
        $ccli = "export webvpn plug-in protocol $_ $storage/$prompt-$date-plugin-$_.jar";
        $occli = $ccli;
        $occli =~ s/^export/import/;
        print "$ccli\n";
        print OUT "$occli\n";
        $obj->send("$ccli\n");
        $obj->expect(15, "$prompt#" );
    }
}

sub url_list {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn url-list\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
```

```

@items = split(/\n+/, $output);

for (@items) {
    chop;
    next if /^Template/ or /show import/ or /^s*$/ or /No bookmarks/;
    $cli="export webvpn url-list $_ $storage/$prompt-$date-urllist-$_.xml";
    $ocli = $cli;
    $ocli =~ s/^export/import/;
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}
}

sub dap {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir dap.xml\n");
    $obj->expect(15, "$prompt#" );

    $output = $obj->before();
    return 0 if($output =~ /Error/);

    $cli="copy /noconfirm dap.xml $storage/$prompt-$date-dap.xml";
    $ocli="copy /noconfirm $storage/$prompt-$date-dap.xml disk0:/dap.xml";
    print "$cli\n";
    print OUT "$ocli\n";
    $obj->send("$cli\n");
    $obj->expect(15, "$prompt#" );
}

sub csd {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("dir sdesktop\n");
    $obj->expect(15, "$prompt#" );
}

```

```

$output = $obj->before();
return 0 if($output =~ /Error/);

$cli="copy /noconfirm sdesktop/data.xml $storage/$prompt-$date-data.xml";
$ocli="copy /noconfirm $storage/$prompt-$date-data.xml disk0:/sdesktop/data.xml";
print "$cli\n";
print OUT "$ocli\n";
$obj->send("$cli\n");
$obj->expect(15, "$prompt#" );
}

sub webcontent {
    $obj = shift;
    $obj->clear_accum();
    $obj->send("show import webvpn webcontent\n");
    $obj->expect(15, "$prompt#" );
    $output = $obj->before();
    @items = split(/\n+/, $output);

    for (@items) {
        s/^\s+//;
        s/\s+$//;
        next if /show import/ or /No custom/;
        next unless (/^\.+s+.$/);
        ($url, $type) = split(/\s+/, $_);
        $turl = $url;
        $turl =~ s/\^+//;
        $turl =~ s/\+\/-/-/;
        $cli = "export webvpn webcontent $url $storage/$prompt-$date-$turl";
        $ocli = $cli;
        $ocli =~ s/^\^export/import/;
        print "$cli\n";
        print OUT "$ocli\n";
        $obj->send("$cli\n");
        $obj->expect(15, "$prompt#" );
    }
}

```

```

sub login {
    $obj = shift;
    $obj->raw_pty(1);
    $obj->log_stdout(0); #turn off console logging.
    $obj->spawn("/usr/bin/ssh $user@$asa") or die "can't spawn ssh\n";
    unless ($obj->expect(15, "password:" )) {
        die "timeout waiting for password:\n";
    }

    $obj->send("$password\n");

    unless ($obj->expect(15, "$prompt>" )) {
        die "timeout waiting for $prompt>\n";
    }
}

sub finish {
    $obj = shift;
    $obj->hard_close();
    print "\n\n";
}

sub restore {
    $obj = shift;
    my $file = shift;
    my $output;
    open(IN,$file) or die "can't open $file\n";
    while (<IN>) {
        $obj->send("$_");
        $obj->expect(15, "$prompt#" );
        $output = $obj->before();
        print "$output\n";
    }
    close(IN);
}

sub process_options {

```

```
if (defined($options{s})) {
    $str= $options{s};
    $storage = "tftp://$str";
}
else {
    print "Enter TFTP host name or IP address:";
    chop($str=<>);
    $storage = "tftp://$str";
}
if (defined($options{h})) {
    $asa = $options{h};
}
else {
    print "Enter ASA host name or IP address:";
    chop($asa=<>);
}

if (defined ($options{u})) {
    $user= $options{u};
}
else {
    print "Enter user name:";
    chop($user=<>);
}

if (defined ($options{w})) {
    $password= $options{w};
}
else {
    print "Enter password:";
    chop($password=<>);
}

if (defined ($options{p})) {
    $prompt= $options{p};
}
else {
    print "Enter ASA prompt:";
    chop($prompt=<>);
}
```

```

}
if (defined ($options{e})) {
    $enable = $options{e};
}
else {
    print "Enter enable password:";
    chop($enable=<>);
}

if (defined ($options{r})) {
    $restore = 1;
    $restore_file = $options{r};
}
}

```

Configuring Auto Update Support

This section includes the following topics:

- [Configuring Communication with an Auto Update Server, page 38-20](#)
- [Configuring Client Updates as an Auto Update Server, page 38-22](#)
- [Viewing Auto Update Status, page 38-23](#)

Auto Update is a protocol specification that allows an Auto Update server to download configurations and software images to many adaptive security appliances, and can provide basic monitoring of the adaptive security appliances from a central location.

The adaptive security appliance can be configured as either a client or a server. As an Auto Update client, the security appliance periodically polls the Auto Update server for updates to software images and configuration files. As an Auto Update server, the security appliance issues updates for adaptive security appliances that have been configured as Auto Update clients.



Note

Auto Update is supported in single context mode only.

Configuring Communication with an Auto Update Server

To configure the adaptive security appliance as an Auto Update client, perform the following steps:

Step 1 To specify the URL of the AUS, enter the following command:

```
hostname(config)# auto-update server url [source interface] [verify-certificate]
```

where *url* has the following syntax:

```
http[s]://[user:password@]server_ip[:port]/pathname
```

SSL is used when **https** is specified. The *user* and *password* arguments of the URL are used for basic authentication when logging in to the server. If you enter the **write terminal**, **show configuration** or **show tech-support** commands to view the configuration, the user and password are replaced with `*****`.

The default port is 80 for HTTP and 443 for HTTPS.

The **source interface** argument specifies which interface to use when sending requests to the AUS. If you specify the same interface specified by the **management-access** command, the Auto Update requests travel over the same IPsec VPN tunnel used for management access.

The **verify-certificate** keyword verifies the certificate that is returned by the AUS.

Step 2 (Optional) To identify the device ID to send when communicating with the AUS, enter the following command:

```
hostname(config)# auto-update device-id {hardware-serial | hostname | ipaddress [if-name]
| mac-address [if-name] | string text}
```

The identifier used is determined by using one of the following parameters:

- *hardware-serial*—The adaptive security appliance serial number.
- **hostname**—The adaptive security appliance hostname.
- **ipaddress**—The IP address of the specified interface. If the interface name is not specified, the IP address of the interface that is used to communicate with the AUS.
- **mac-address**—The MAC address of the specified interface. If the interface name is not specified, the MAC address of the interface that is used to communicate with the AUS.
- **string**— The specified text identifier, which cannot contain white space or the characters ‘, “, , >, & and ?.

Step 3 (Optional) To specify how often to poll the AUS for configuration or image updates, enter the following command:

```
hostname(config)# auto-update poll-period poll-period [retry-count [retry-period]]
```

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is 0.

The *retry-period* argument specifies how long to wait in minutes between retries. The default is 5 minutes.

Step 4 (Optional) To schedule a specific time for the security appliance to poll the Auto Update server, enter the following command:

```
hostname(config)# auto-update poll-at days-of-the-week time [randomize minutes] [retry_count
[retry_period]]
```

The *days-of-the-week* argument is any single day or combination of days: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday. Other possible values are *daily* (Monday through Sunday), *weekdays* (Monday through Friday), and *weekends* (Saturday and Sunday).

The *time* argument specifies the time in the format HH:MM at which to start the poll. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

The **randomize** *minutes* argument specifies the period to randomize the poll time following the specified start time. The range is from 1 - 1439 minutes.

The *retry_count* argument specifies how many times to try reconnecting to the Auto Update Server if the first attempt fails. The default is 0.

The *retry_period* argument specifies how long to wait between connection attempts. The default is 5 minutes. The range is from 1 - 35791 minutes.

- Step 5** (Optional) If the Auto Update Server has not been contacted for a certain period of time, to stop passing traffic, enter the following command:

```
hostname(config)# auto-update timeout period
```

where *period* specifies the timeout period in minutes, which ranges from 1 - 35791. The default is not to time out (0). To restore the default, enter the **no** form of this command.

Use this command to ensure that the adaptive security appliance has the most recent image and configuration. This condition is reported with system log message 201008.

In the following example, a adaptive security appliance is configured to poll an AUS with IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

The security appliance is also configured to use the hostname as the device ID and to poll every Friday and Saturday night at a random time between 10:00 p.m. and 11:00 p.m. After a failed polling attempt, the security appliance tries to reconnect to the AUS 10 times, and waits three minutes between attempts.

```
hostname(config)# auto-update server https://juser:example@209.165.200.224:1742/management
source outside verify-certificate
hostname(config)# auto-update device-id hostname
hostname(config)# auto-update poll-at Friday Saturday 22:00 randomize 60 2 10
```

Configuring Client Updates as an Auto Update Server

The **client-update** command lets you enable the update for adaptive security appliances configured as Auto Update clients. This command lets you specify the type of software component (ASDM or boot image), the type or family of adaptive security appliance, revision numbers to which the update applies, and a URL or IP address from which to obtain the update.

To configure the adaptive security appliance as an Auto Update server, perform the following steps:

-
- Step 1** In global configuration mode, enable client updates by entering the following command:

```
hostname(config)# client-update enable
hostname(config)#
```

- Step 2** Configure the parameters for the client updates that you want to apply for the adaptive security appliances using the **client-update** command:

```
client-update {component {asdm | image} | device-id dev_string |
family family_name | type type} url url-string rev-nums rev-nums}
```

The **component** {**asdm** | **image**} keyword specifies the software component (ASDM or boot image) of the adaptive security appliance.

The **device-id** *dev_string* argument specifies a unique string that the Auto Update client uses to identify itself. The maximum length is 63 characters.

The **family** *family_name* argument specifies the family name that the Auto Update client uses to identify itself. It can be asa, pix, or a text string with a maximum length of 7 characters.

The **rev-nums** *rev-nums* argument specifies the software or firmware images for this client. Enter up to 4, in any order, separated by commas.

The **type** *type* argument specifies the type of clients to notify. Because this command is also used to update Windows clients, the list of clients includes several Windows operating systems. The adaptive security appliance to notify is the ASA 5580.

The **url** *url-string* argument specifies the URL for the software/firmware image. This URL must point to a file appropriate for this client. For all Auto Update clients, you must use the protocol “http://” or “https://” as the URL prefix.

- Step 3** Configure the parameters for the client update that you want to apply to all adaptive security appliances of a particular type. That is, specify the type of adaptive security appliance and the URL or IP address from which to obtain the updated image. In addition, you must specify a revision number. If the revision number of the remote adaptive security appliance matches one of the specified revision numbers, the client ignores the update.

The following example configures a client update for ASA 5580 adaptive security appliances:

```
hostname(config)# client-update type asa5580 component asdm url  
http://192.168.1.114/aus/asdm611.bin rev-nums 8.1(1)
```

Viewing Auto Update Status

To view the Auto Update status, enter the following command:

```
hostname(config)# show auto-update
```

The following is sample output from the **show auto-update** command:

```
hostname(config)# show auto-update  
Server: https://*****@209.165.200.224:1742/management.cgi?1276  
Certificate will be verified  
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes  
Timeout: none  
Device ID: host name [corporate]
```

