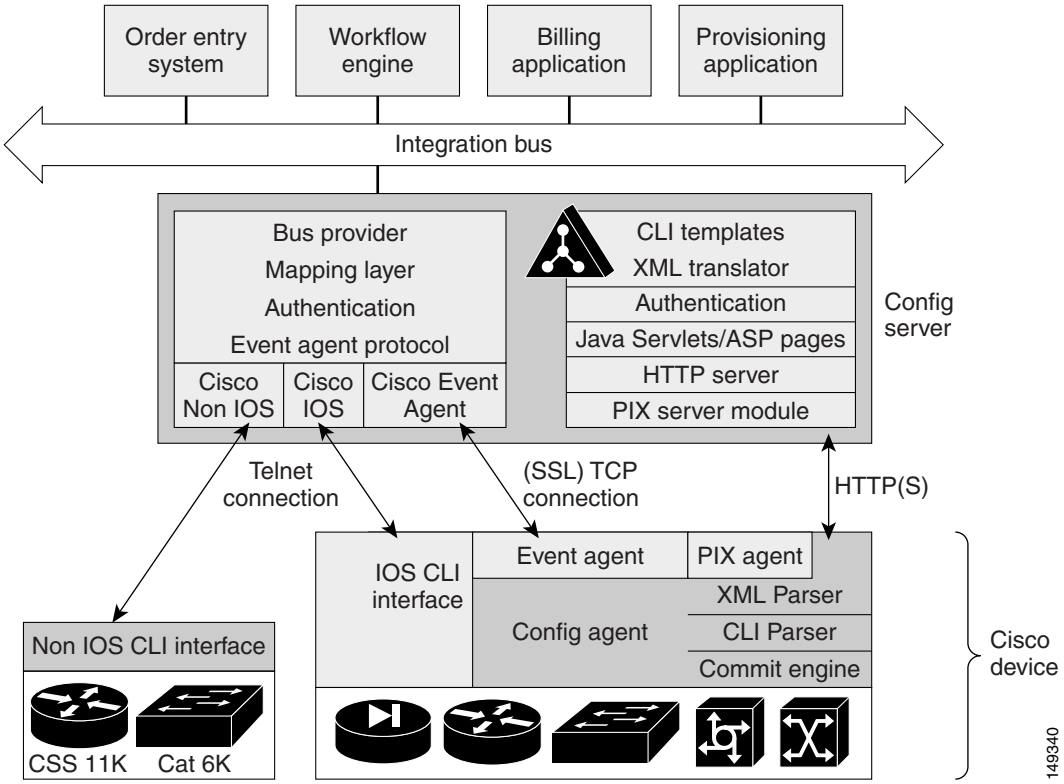
CHAPTER 21

# PIX Firewall Device Support

Cisco Configuration Engine provides configuration management and image service to Cisco PIX firewall devices (PIX device). Figure 21-1 shows a functional block diagram of Cisco Configuration Engine including the PIX device interface module.

***Figure 21-1 PIX-Compatible Configuration Engine Module Interaction***




**Note** Encryption must be enabled during setup for PIX devices to supported by Cisco Configuration Engine.

# PIX Device Polls for Updates

The PIX device contacts the PIX module in the Cisco Configuration Engine to report information about itself. This occurs when the PIX starts, when any of the reported information changes, and whenever the PIX wants to check for updates. PIX sends the **DeviceDetails** message to the server. The **DeviceDetails** message provides the Cisco Configuration Engine an update of the version of software that the device is currently running. The information received in **DeviceDetails** is logged into the log file (*pix.log*) for reference.

The server responds with the **UpdateInfo** message. This message contains (optionally):

- Checksum and URL for the configuration file the PIX should be running
- Checksum and URL for the PIX image
- Checksum and URL for the PIX Device Manager (PDM) image
- URL for reporting any errors

The PIX compares the checksum in the message with the current checksum of the component concerned. In the case of configuration, it also calculates the cryptochecksum of the running configuration and compare that with the one calculated the last time when the configuration was updated from the Cisco Configuration Engine. An update is required if the checksum (or cryptochecksum) differs.

If a software/configuration update is required, the PIX sends requests on the respective URLs.

## Configuration Processing

For any configuration update that is required, the PIX sends an HTTPS GET request to the returned URL. The configuration file is completely read into a local buffer before being applied. This is to prevent a connection error from leaving the PIX in a partially configured state. If there are no errors (or the *errors* attribute of the **config-data** message is *continue*) while applying the configuration commands, then the running configuration is copied to flash with the **write memory** command. All configuration files work in the *replace* mode.

Completion of configuration download by a PIX device results in a log file entry indicating the same in *pix.log*.

**Note** The log entry does not mean that the configuration has been successfully applied on a PIX device. It only means that the PIX device has downloaded the configuration file.

## Image Processing

The **DeviceDetails** XML sent along with the initial HTTPS POST optionally has information regarding the PIX image, its version and checksum. Cisco Configuration Engine returns with the UpdateInfo XML containing image URLs and checksums based on the entries in the directory. The PIX downloads and applies images one after the other (and reload itself if required). Any error is processed as mentioned below.

**Note** There is no notification of successful image download because image distribution might be external to Cisco Configuration Engine and hence the PIX server cannot keep track of the same. Also, the PIX device does not provide any image upgrade successful indication.

# Error Processing

All errors are reported by way of HTTPS POST to the error URL using the **ErrorList** message.

Each configuration error report (type=error, warning or info) is logged by the Cisco Configuration Engine into *pix.log*. The log file is cyclic to limit disk space usage.

**Note** An error occurring during configuration does not mean that the downloaded configuration is not been applied on the PIX entirely. It only means that the error mentioned in the log file has happened with respect to this particular device.

Any error or notification (type= warning, notification, informational, debugging, emergency, alert, critical and error) that occurs while retrieving the data at one of the URLs received from the Cisco Configuration Engine results in log file entries.

If a failure is encountered during the processing of any of the URLs in the UpdateInfo response from the server, the error is reported to the Error URL. Also, processing of all URLs received in the current call home is discontinued. Any further processing is deferred till the PIX calls home again.

After all the updates are successfully completed, another **DeviceDetails** message is sent to the Cisco Configuration Engine by the PIX device. Cisco Configuration Engine again sends the **UpdateInfo** and checksum. The PIX device compares the checksums and finds that no further updates are required.

# Processing a DeviceDetails Request from PIX Device

The sequence of processing a DeviceDetails request from a PIX device is as follows:

1. PIX device contacts the Cisco Configuration Engine with **DeviceDetails** as XML payload by means of an HTTPS post request.
2. New PIX Configuration servlet receives request, parses XML, and retrieves DeviceID.
3. Device is authenticated.
4. Template associated with this DeviceID is processed to generate a configuration file.
5. Cnfiguration file is converted into XML format as per the PIX DTD and the file is saved (over-written in case a file is already present for this DeviceID).
6. Checksum of XML configuration file is calculated and URL noted.
7. URLs and checksums for pix image and PDM images are retrieved from image object attached with the PIX device.
8. Checksums and URLs for configuration file and various images (if the corresponding checksum differs) and the Error URL are sent to the PIX device as an HTTP response with an XML payload (UpdateInfo).
9. Device now requests for configuration/image based on the content of the UpdateInfo response.
10. If errors are encountered, information is posted to error URL.
11. Error servlet logs the errors to *pix.log*.

## PIX DeviceID

The following PIX CLI decides the value of DeviceID sent by PIX in the DeviceDetails request:

**[no] auto-update device-id hardware-serial | hostname | ipaddress [***if-name***] | mac-address [***if-name***] | string text**

- **auto-update device-id** command specifies the device ID to send when polling the Management server.
- **no auto-update device-id** command resets the device ID to the default of hostname.
- **hardware-serial** option uses the PIX serial number.
- **hostname** option uses the PIX host name.
- **ipaddress** option uses the IP address of the interface with the name **if-name**.

  If the interface name is not specified, it uses the IP address of the interface used to communicate with the remote management server.
- **mac-address** option uses the MAC address of the interface with the name *if-name*.

  If the interface name is not specified, it uses the MAC address of the interface used to communicate with the remote management server.
- **string** option uses the specified *text*.

  The text can not contain white space or the characters ‘, “, <, >, & and ?.

**Note** Since the DeviceID provided by the PIX is internally mapped to ConfigID and EventID in the Cisco Configuration Engine, it only supports hyphen (-), underscore (_), period (.) and alphanumeric characters.

# Security Considerations

Since PIX devices are firewall devices and configuration information is vital, transport of this information is made secure by the use of SSL.

HTTPS has been enforced as the transport protocol between PIX devices and Cisco Configuration Engine under all circumstances. **DeviceDetails**, **Update Info**, **ErrorInfo** and configuration files are transported only using HTTPS. The authorization mechanism used in Configuration Service has been leveraged in the PIX server module. The URLs supplied by you towards PDM/pix-image can use HTTP or HTTPS.

# PIX Device Polling Setup

PIX devices can be configured to poll the Cisco Configuration Engine at regular intervals for configuration or image updates. This entry has to be made by you on the PIX device itself. Details are available from PIX device documentation. CLI format for the same is as follows:

**Usage: auto-update device-id hardware-serial | hostname |**

**ipaddress [<if_name>] | mac-address [<if_name>] | string <text>**

**no auto-update device-id**

**auto-update poll-period <poll-period> [<retry-count>**

**[<retry-period>]]**

**no auto-update poll-period**

**auto-update server <url> [verify-certificate]**

**no auto-update server**

**auto-update timeout <period>**

**no auto-update timeout**

Example:

```
auto-update device-id string myPIXDevice
auto-update poll-period 120
auto-update server https://********@cns-ie2100/cns/PIXConfig
```

The URI to be polled on the Cisco Configuration Engine is:

**/cns/PIXConfig**

The **auto-update poll-period** command specifies how often to poll the Management server for configuration or image updates. The *poll-period* parameter specifies how often (in minutes) to check for an update. The default is 720 (12 hours). The *retry-count* option specifies how many times to try re-connecting to the server if the first attempt fails. The default is 0. The *retry-period* option specifies how long to wait (in minutes) between retries. The default is 5.

The **no auto-update poll-period** command resets the poll period to the default.

Also, you must to map the hostname of the server on the PIX device with its IP address. You can do this by using the *name* command as follows:

pixfirewall# **conf t**

pixfirewall(config)# **name <ip_address of the server> <hostname of the server>**

# Configuration and Restrictions

PIX compatibility module is set up along with Configuration Service during the initial setup of the system. You need not do anything specifically to enable PIX compatibility.

PIX devices with **software versions of 6.2.1 and higher** are supported by Cisco Configuration Engine (auto-update from PIX device side was introduced in this version). All PIX hardware platforms that run software version 6.2.1 or higher will be supported.

The configuration files will be generated with options config-action= **replace** and errors=**revert**. No other options are supported.