CHAPTER 1

Understanding Host Input

The FireSIGHT System provides two tools for importing data from other sources on your network to augment your network map: the host input API and the host input import tool.

If your organization has the expertise to create Perl scripts, the host input API allows you to script direct data transfer between a third-party application and the network map. For example, you might have a patch management application on your network that contains information about the current patch levels for the hosts on your network. You could import the third-party fix information for each host into the network map. If you set up a map of the names that the third-party application uses for each patch and invoke it before adding the fixes, the system can use that information to update the vulnerability list on each host to deactivate vulnerabilities addressed by the fix. The host input API allows you to create a script that maps third-party data structures to Cisco data structures, so you can re-run the script to import new data as needed, as long as the names of data elements do not change on either side.

If you do not have a programmer available to you, or if you want to import a set of data and do not need to re-run similar imports in the future, you can create a text file containing the data and use the host input import tool to perform the import on the Defense Center using the `nmimport.pl` script.

For example, if you are setting up a new installation of FireSIGHT, you might want to make sure that all the computers listed in your asset management software exist in the network map. You could export the host data from the asset management application, format the results into an appropriately formatted text file, and import the host data using the host input import tool. If the asset management system includes operating system information for each host, you could set up a third-party product map for the asset management system and map each third-party operating system label to the corresponding Cisco label. You can set that map before you run the import, and the system will associate the appropriate Cisco operating system definition with each host.

There are five major steps to setting up a host input API connection with the FireSIGHT System:

1. If you want to perform impact correlation using third-party host data, you can configure third-party product maps to map service, operating system, or fix definitions to Cisco product or fix definitions, using the Defense Center web interface.

2. If you want to import third-party vulnerabilities, you can configure third-party vulnerability maps to map third-party vulnerability identification strings to Cisco vulnerability IDs, using the Defense Center web interface. Note that you can also perform this mapping in your client using the SetCurrent3rdPartyMap API function with the appropriate vulnerability keys.

3. Write a script that imports data to hosts in the network map using the host input API, including calls to invoke third-party product maps as needed.

4. Log in as `admin` on your Defense Center.

5. Run the script to import the data.

There are five major steps to using the host input import tool with the FireSIGHT System:

1. If you want to perform impact correlation using third-party host data, you can configure third-party product maps to map service, operating system, or fix definitions to Cisco product or fix definitions, using the Defense Center web interface.
2. If you want to import third-party vulnerabilities, you can configure third-party vulnerability maps to map third-party vulnerability identification strings to Cisco vulnerability IDs, using the Defense Center web interface. Note that you can also perform this mapping in your import file.
3. Export data from a third-party application and format it to match the formatting guidelines provided in Using the Host Input Import Tool, page 3-1.
4. Run the import tool.
5. To use the import tool, log in as `admin` on your Defense Center. Use the import tool to set the third-party product map. Use the import tool to import data from the import file you created.

Prerequisites

To understand the information in this guide, you should be familiar with the features and nomenclature of the FireSIGHT System and the function of its components (in particular, the network map), and with the various related event data the system generates. Information about these functions, together with definitions of unfamiliar or product-specific terms, may be obtained from the *FireSIGHT System User Guide*. Additional information about the data fields documented in this guide may be obtained from the *User Guide* as well.

Product Version Compatibility

The following table describes the product version required for various host input functionality:

Table 1-1 Product Version Compatibility

Functionality	Product Version
Host input functionality	FireSIGHT System version 4.9+
`ScanUpdate` function, `AddHostAttribute` function, `DeleteHostAttribute` function, `SetSourceType` function, `DeleteScanResult` function, `AddScanResult` function, `ScanFlush` function, setting source types, setting source type IDS	FireSIGHT System version 4.10+
Host input external client functionality	FireSIGHT System version 5.0+
Host input mobile device identification functionality	FireSIGHT System version 5.1+
`DeleteClientApp` function	FireSIGHT System version 5.1.1+
IPv6 address support	FireSIGHT System version 5.2+

Document Conventions

The following table lists the names used in this book to describe the various data field formats employed in host input calls. Numeric constants used by the host input API or host input import tool are typically unsigned integer values. Bit fields use low order bits unless otherwise noted. For example, in a one-byte field that contains five bits of flag data, the low order five bits will contain the data.

Table 1-2 Key Value Data Type Conventions

Data Type	Description
uint	Unsigned integer
uint8	Unsigned 8-bit integer
uint32	Unsigned 32-bit integer
string	Variable length bytes containing character data.
[n]	Array subscript following any of the above data types to indicate n instances of the indicated data type, for example, uint8[4] is an array of four 8-bit elements.
variable	Collection of various data types.

Host Input Scripting Resources

The following describes some of the topics explained in the documentation and where to look for more information.

Table 3 Host Input Resources

To learn more about...	Look in...
the host input API	Using the Host Input API, page 2-1
the host input import tool	Using the Host Input Import Tool, page 3-1
writing a script to connect to the host input API	Writing Host Input API Scripts, page 2-1
writing a script to import data using the host input API	Running a Host Input API Script, page 2-2
calling a specific host input API function	Host Input API Functions, page 2-5
guidelines for writing an import file to use with the host input import tool	Writing Host Input Import Files, page 3-3
syntax for a specific host input function to include in an import file	Host Input Import Syntax, page 3-6
running the host input import tool	Running a Host Input Import, page 3-29
installing, configuring, and running the host input reference client	Using the Host Input Reference Client, page 4-2