CHAPTER 12

Using Access Control

This chapter describes how to configure and maintain access control lists. Access control lists are used to permit or deny access to the LightStream 1010 ATM switch.

The access control list is used by the Asynchronous Transfer Mode (ATM) signaling software to filter setup messages on an interface or subinterface as either destination or source. Access lists can be used to deny connections that are known to be security risks and permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, denying access to security risks offers more control.

The *LightStream 1010 ATM Switch Command Reference* publication provides the complete syntax for every switch configuration command and describes the **no** form of each command.

During initial configuration perform the following steps to use access control to filter setup messages:

Step 1 Create template alias. This allows you to use real names instead of ATM addresses in your ATM filter expressions.

Step 2 Create the ATM filter expression.

Step 3 Create the ATM access group either globally or on a specific interface.

Step 4 Confirm the configuration.

The following sections describe access control configuration, including examples:

- Configure Global Template Alias
- Configure Global ATM Access Control
- Configure ATM Address Pattern-Match Expression
- Configure ATM Interface Access Control
- Filter IP Packets at the IP Interfaces

Configure Global Template Alias

Configure a global ATM template alias using the following commands using the **no** form of the command to delete the specified alias:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Configure a global ATM address template alias.	**atm template-alias** *name template*

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

Syntax:

name —The name for the template alias.

template—An ATM address template which may be a single ATM address that matches itself or contain wildcards and/or a prefix or suffix, that allows a single template to match many addresses.

The symbols used for wildcards and prefix/suffix are as follows:

- An asterisk (*) to match any single 4-bit nibble in the address.
- An ellipsis (...) to match any number of leading or trailing 4-bit hexadecimal digits in the address.
- An asterisk (*) to match any single binary digit in a 4-bit nibble in the address, where the four binary bits are enclosed within parentheses.

Examples:

The following example creates a template alias named *training* with the ATM address template 47.1328 and using ellipse (...) to fill in the trailing 4-bit hexadecimal digits in the address:

```
Switch(config)#atm template-alias training 47.1328...
```

The following example creates a template alias named *competition* with the ATM address template 47.0012. plus any additional addresses matching (10**) and using the ellipse:

```
Switch(config)#atm template-alias competition 47.0012.(10**)...
```

The following example creates a template alias named *bad_users* with the ATM address template ending with 1234. and the binary digits (01*1):

```
Switch(config)#atm template-alias bad_users...1234.(01*1)
```

Configure Global ATM Access Control

To create an ATM filter for the entire switch use the global address pattern-matching filter. Using the **no** form of the command deletes the specified ATM filter set:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Configure a global ATM address filter set.	**atm filter-set** *name* [**permit**|**deny**] *template*

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

If neither **permit** nor **deny** is specified, **permit** is assumed. If an address does not match any of the filter set entries, an implicit *deny* is returned as the permit or deny action of the filter set.

Examples:

The following example creates a global filter named filter_1 that permits access to the specific ATM address 47.0000.8100.1234.0003.c386.b301.0003.c386.b301.00:

```
Switch(config)#atm filter-set filter_1 permit 47.0000.8100.1234.0003.c386.b301.0003.c386.b301.00
```

The following example creates a global filter named filter_2 that denies access to the specific ATM address 47.000.8100.5678.0003.c386.b301.0003.c386.b301.00:

```
Switch(config)#atm filter-set filter_2 deny 47.0000.8100.5678.0003.c386.b301.0003.c386.b301.00
```

The following example creates a global filter named filter_3 that denies access to all ATM addresses that begin with the prefix 47.840F:

```
Switch(config)#atm filter-set filter_3 deny 47.840F...
```

The following example creates a global filter named filter_4 that denies access to all ATM addresses described by the ATM template alias bad_users:

```
Switch(config)#atm filter-set filter_4 deny bad_users
```

Configure ATM Address Pattern-Match Expression

The following commands create global ATM address pattern-matching filter expressions:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Define a simple filter expression that is pattern matched only if the pattern given by *term* matches.	**atm filter-expr** *name term*
Define a filter expression that is pattern matched only if the pattern given by *term* is *not* matches.	**atm filter-expr** *name* **not** *term*
Define a filter expression that is pattern matched only if *both* of the patterns given by the two *terms* matches.	**atm filter-expr** *name* term **or** *term*
Define a filter expression that is pattern matched only if *one* of the patterns, but *not* both, given by the two *terms* matches.	**atm filter-expr** *name* term **xor** *term*
Define a filter expression using logical operators **or, and**, and **xor**.	**[no] atm filter-expr** *name*

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

Syntax:

If neither **permit** nor **deny** is specified, **permit** is assumed. If an address does not match any of the filter set entries, an implicit *deny* is returned as the permit or deny action of the filter set.

name—the name of the pattern-matching filter expression.

term—can be any of the following:

- A previously defined address pattern-matching expression.
- Source *filter-set name*: filter applied to calling party address.
- Destination *filter-set name*: filter applied to called party address.

For commands with two *term*s the evaluation sequence is from left to right of the expression, for example, commands using logical operators **or, and**, and x**or**.

For commands using logical operators **or** plus **and**, the evaluation for the second *term* is conducted only when necessary. For example, the evaluation for the second *term* is omitted if the truth or falsehood can already be concluded from the evaluation for the first *term*.

Examples:

The following example defines a simple filter expression that is pattern-matched only if the pattern given by *term* filter_1 is matched:

```
Switch(config)#atm filter-expr training filter_1
```

The following example defines a filter expression that is pattern-matched only if the pattern given by *term* filter_1 is *not* matched.

```
Switch(config)#atm filter-expr training not filter_1
```

The third form defines a filter expression that is pattern-matched if *either* of the patterns given by the two *terms* filter_1 **and** filter_2 are matched.

```
Switch(config)#atm filter-expr training filter_2 or filter_1
```

The following example defines a filter expression that is pattern-matched only if *both* of the patterns given by the two *terms* are matched.

```
Switch(config)#atm filter-expr training filter_1 and source filter_2
```

The following example defines a filter expression that is pattern-matched only if *one* of the patterns, but *not* both, given by the two *terms* are matched.

```
Switch(config)#atm filter-expr training filter_2 xor filter_1 and destination filter_2
```

Configure ATM Interface Access Control

The command to subscribe an ATM interface or subinterface to an existing ATM address pattern-matching filter expression is as follows using the **no** form of the command to delete an address access filter subscription on a specified interface or subinterface:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Select the interface or subinterface to be configured.	**interface atm** *card/sub_card/port* [*.vpt #*]
Configure an existing ATM address pattern matching filter expression.	**atm access-group** *name* [**in** | **out**]

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

Syntax:

name—The name of the filter expression or filter set.

in—Indicates the filter should be applied to incoming SETUP message.

out—Indicates the filter should be applied to outgoing SETUP message. This is the default.

If neither **in** nor **out** is specified, the filter is applied to outgoing SETUP message.

Example:

The following example defines ATM template alias training to be filtered out:

```
Switch(config)#interface atm 3/0/0
Switch(config-if)#atm access-group training out
```

The following example defines ATM template alias marketing to be allowed in:

```
Switch(config)#interface atm 3/0/0
Switch(config-if)#atm access-group training out
Switch(config-if)#atm access-group marketing in
```

Display ATM Filter Configuration

Exec commands to display access control configuration are as follows:

Task	Command
Display a summary of ATM filter set.	**show atm filter-set** [*name*]
Display a specific or a summary of ATM filter expression.	**show atm filter-expr** [*name*]

Examples:

The following command displays the configured ATM filters:

```
Switch#show atm filter-set
ATM filter set filter_1
  permit 47.0000.8100.1234.0003.c386.b301.0003.c386.b301.0
Switch#
```

The following command displays the configured ATM filter expressions:

```
Switch#show atm filter-expr
training = dest filter_1
Switch#
```

Filter IP Packets at the IP Interfaces

Internet Protocol (IP) packet filtering helps control packet movement through the network. Such control can help limit network traffic and restrict network use by certain users or devices. To permit or deny packets from crossing specified IP interfaces, we provide access lists.

You can use access lists in several ways:

- To control the transmission of packets on an IP interface
- To control virtual terminal line access
- To restrict contents of routing updates

This section summarizes how to create IP access lists and how to apply them.

Note This section applies to the IP interfaces only.

An access list is a sequential collection of permit and deny conditions that apply to IP addresses. The LightStream 1010 software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two steps involved in using access lists are as follows:

Step 1 Create an access list by specifying an access list number and access conditions.

Step 2 Apply the access list to interfaces or terminal lines.

These steps are described in the next sections.

Create Standard and Extended IP Access Lists

The software supports three styles of access lists for IP interfaces:

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses for matching operations, as well as optional protocol type information for finer granularity of control.
- Dynamic extended IP access lists grant access per user to a specific source or destination host through a user authentication process. In essence, you can allow user access through a firewall dynamically, without compromising security restrictions.

To create a standard access list, perform one of the following tasks in global configuration mode:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Define a standard IP access list using a source address and wildcard.	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
Define a standard IP access list using an abbreviation for the source and source mask of 0.0.0.0 255.255.255.255.	**access-list** *access-list-number* {**deny** | **permit**} **any**

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

To create an extended access list, perform one of the following tasks in global configuration mode:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Define an extended IP access list number and the access conditions. Use the **log** keyword to get access list logging messages, including violations.	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
Define an extended IP access list using an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255 and an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255.	**access-list** *access-list-number* {**deny** | **permit**} *protocol* **any any**
Define an extended IP access list using an abbreviation for a source and source wildcard of *source* 0.0.0.0 and an abbreviation for a destination and destination wildcard of *destination* 0.0.0.0.	**access-list** *access-list-number* {**deny** | **permit**} *protocol* **host** *source* **host** *destination*
Define a dynamic access list.	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

After an access list is created initially, any subsequent additions (possibly entered from the terminal) are placed at the end of the list. In other words, you cannot selectively add or remove access list command lines from a specific access list.

Note Keep in mind when making the standard and extended access list by default, the end of the access list contains an implicit deny statement for everything if it does not find a match before reaching the end. Further, with standard access lists, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

Apply an IP Access List to an Interface or Terminal Line

After an access list is created, you can apply it to one or more interfaces. Access lists can be applied on *either* outbound or inbound interfaces. The following two tables show how this task is accomplished for both terminal lines and network interfaces.

Perform the following task in line configuration mode:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Select the line to be configured.	**line** [**aux**|**console**|**vty**] *0*
Restrict incoming and outgoing connections between a particular virtual terminal line (into a device) and the addresses in an access list.	**access-class** *access-list-number* {**in** | **out**}

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

Perform the following task in interface configuration mode:

Task	Command
At the privileged EXEC prompt, enter configuration mode from the terminal.	**configure**[1] [**terminal**]
Select the interface or subinterface to be configured.	**interface atm** *card/sub_card/port*
Control access to an interface.	**ip access-group** *access-list-number* {**in** | **out**}

1. This command is documented in the *LightStream 1010 ATM Switch Command Reference* publication.

For inbound access lists, after receiving a packet, the LightStream 1010 software checks the source address of the packet against the access list. If the access list permits the address, the software continues to process the packet. If the access list rejects the address, the software discards the packet and returns an Internet Control Message Protocol (ICMP) Host Unreachable message.

For outbound access lists, after receiving and routing a packet to a controlled interface, the software checks the source address of the packet against the access list. If the access list permits the address, the software transmits the packet. If the access list rejects the address, the software discards the packet and returns an ICMP Host Unreachable message.

When you apply an access list (standard or extended) that has not yet been defined to an interface, the software will act as if the access list has not been applied to the interface and will accept all packets. Remember this behavior if you use undefined access lists as a means of security in your network.

Note Set identical restrictions on all the virtual terminal lines, because a user can attempt to connect to any of them.

IP Access List Examples

In the following example, network 36.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 36.0.0.0 address specify a particular host. Using access list 2, the LightStream 1010 software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 36.0.0.0 subnets.

```
access-list 2 permit 36.48.0.3
access-list 2 deny 36.48.0.0  0.0.255.255
access-list 2 permit 36.0.0.0  0.255.255.255
interface ethernet 0
ip access-group 2 in
```

Examples of Implicit Masks in IP Access Lists

IP access lists contain *implicit* masks. For instance, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask. Consider the following example configuration:

```
access-list 1 permit 0.0.0.0
access-list 1 permit 131.108.0.0
access-list 1 deny 0.0.0.0 255.255.255.255
```

For this example, the following masks are implied in the first two lines:

```
access-list 1 permit 0.0.0.0 0.0.0.0
access-list 1 permit 131.108.0.0 0.0.0.0
```

The last line in the configuration (using the deny keyword) can be left off, because IP access lists implicitly *deny* all other access. This is equivalent to finishing the access list with the following command statement:

```
access-list 1 deny 0.0.0.0 255.255.255.255
```

The following access list only allows access for those hosts on the three specified networks. It assumes that subnetting is not used; the masks apply to the host portions of the network addresses. Any hosts with a source address that does not match the access list statements is rejected.

```
access-list 1 permit 192.5.34.0  0.0.0.255
access-list 1 permit 128.88.0.0  0.0.255.255
access-list 1 permit 36.0.0.0  0.255.255.255
! (Note: all other access implicitly denied)
```

To specify a large number of individual addresses more easily, you can omit the address mask that is all zeros from the **access-list** global configuration command. Thus, the following two configuration commands are identical in effect:

```
access-list 2 permit 36.48.0.3
access-list 2 permit 36.48.0.3  0.0.0.0
```

Examples of Configuring Extended IP Access Lists

In the following example, the first line permits any incoming Transmission Control Protocol (TCP) connections with destination ports greater than 1023. The second line permits incoming TCP connections to the simple mail transfer protocol (SMTP) port of host 128.88.1.2. The last line permits incoming ICMP messages for error feedback.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 gt 1023
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
access-list 102 permit icmp 0.0.0.0 255.255.255.255 128.88.0.0 255.255.255.255
interface ethernet 0
ip access-group 102 in
```

For another example of using an extended access list, suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the Ethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the switch will always be accepting mail connections on port 25 is what makes it possible to separately control incoming and outgoing services.The access list can be configured on either the outbound or inbound interface.

In the following example, the Ethernet network is a Class B network with the address 128.88.0.0, and the mail host's address is 128.88.1.2. The keyword **established** is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the acknowledgment (ACK) or RST bits set, which indicate that the packet belongs to an existing connection.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 established
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
interface ethernet 0
ip access-group 102 in
```