INTRODUCTION
Using time-based access lists, one can control the time one or more access list statements go into effect. This capability enables the router administrator to configure a common access list that takes into consideration preplanned changes in an organization’s security policy, providing much needed flexibility to enhance the communications of an organization. For example, through the use of time-based access lists, an organization could restrict access to certain servers to predefined time periods or could even specify which TCP/IP applications could be supported by one or more hosts during certain periods of time.

OVERVIEW
This article represents the second in a series of articles focused on new developments in Cisco router access list technology. Because the router represents the point of entry into a network, the ability to perform different types of packet filtering enables organizations to use access lists as a mechanism to put into effect different types of security policies. This article focuses attention on time-based access lists.

PAYOFF IDEA
Until recently, it was difficult for a router administrator to alter an access list at a particular point in time. To accomplish this task required the router administrator to delete one access list and create and apply a second. Although the administrator could perform this task remotely, it was not a pleasant thought to consider having to access a PC to establish a connection to a corporate router at midnight on a Saturday evening or at another nonstandard time to put a new organizational security policy into effect. In addition, because many organizations can require periodic changes to their security policies based on different periods of time that reflect additions and removals of equipment attached to their networks, this activity could cause networking personnel to work extended hours. Perhaps recognizing such problems, Cisco Systems introduced time-based access lists in Version 12.0 of its Internetworking Operating System (IOS).
In actuality, time-based access lists represent a feature added to Cisco router access lists and do not actually represent a separate type of access list. However, under Cisco terminology this feature has the label "time-based access lists" and this label is used in this article to refer to the operation and utilization of this access list feature.

**PROCEDURE**

The basic procedure necessary to implement a time-based access list is a relatively simple two-step process. First, one defines a time range. Once that is accomplished one can reference the time range in the access list through the use of the keyword *time-range*.

Defining a time range also represents a two-step process. First, use the *time-range* command to identify the time range by an appropriate name. Next, use an *absolute* or one or more *periodic* statements, to define when the named time range is to occur. The format associated with each IOS command is shown below:

\[
\text{time-range} \ \text{time-range-name} \\
\text{absolute} \ [\text{start time date}] [\text{end time date}] \\
\text{periodic} \ \text{days-of-the-week} \ \text{hh:mm} [\text{days-of-the-week}] \ \text{hh:mm}
\]

where:

- **time-range-name**: Represents the name one wants to use to identify the time-range for reference in an access list.
- **time**: Represents the time entered with respect to a 24-hour clock, followed by minutes (hh:mm).
- **date**: Represents the date expressed in the format day month year.
- **days-of-the-week**: Represents the day or days the associated time range is to be into effect. Arguments can be a single day (i.e., Monday) or multiple days (i.e., Monday Wednesday Friday).

To obtain an appreciation for the use of the time-range command and the use of different variables in the absolute and periodic statements, one can examine the command and each statement in detail.

**THE TIME-RANGE COMMAND**

The *time-range* command is used to associate a name to the time range one will create using one absolute and one or more periodic statements. That is, the *time-range* command only supports one absolute statement. However, the time-range command supports multiple periodic statements, which are optional. To illustrate the use of the *time-range* command, assume that one wants to define a period during which HTTP
traffic will be allowed. Although one can use any name to reference the
time range to be created, an appropriate name would be allow-http.
Thus, assuming one has selected allow-http as the time range name, one
would use the time-range command as follows:

    time-range allow-http

Once the time-range command has been used and a name to identify
the range has been assigned, one would then use an absolute a or peri-
odic statement to define the time range that will be associated with the
name. Proceeding further, one can now focus on the use of each of those
statements in a time-based access list.

THE ABSOLUTE STATEMENT

The absolute statement is used to specify an absolute time when the time
range will be in effect. As previously noted in examining the basic format
of this statement, the keyword absolute is followed by the keyword start
and the keyword end, with both start and end followed by the starting
and ending time one wants any associated permit or deny statements in
the access list to go into effect. In specifying the time, it is important to
remember that it is specified in military fashion, with respect to a 24-hour
format, while the date is expressed in the format of day, month, year. For
example, 3:00 p.m. would be specified as 15:00, while December 17,
2000 would be specified as 17 12 2000. For non-Europeans it is easy to
enter the incorrect date, as North Americans are normally accustomed to
the format in which the month is followed by the day and year. Thus, it
is important to verify that the correct format has been used to specify the
date. It is also important to note the effect resulting from failing to enter
a start or end date because both are optional. If one selects to eliminate
either a start or end date, the permit or deny statement associated with
the time-based access list will either go into effect immediately if no start
date is specified, or will result in an indefinite time period if no end date
is specified. Because either or both situations may not represent one's in-
tention, this illustrates the importance associated with carefully consid-
ering the format of each statement and the manner by which variables
associated with a statement are entered.

To illustrate the use of the absolute statement, assume that one only
wants to enable HTTP traffic from 6:00 p.m. until 7:00 a.m. One would
assign that time range with the following absolute statement:

    absolute start 18:00 end 07:00

Note that in the preceding example, the time range occurs from 6:00 p.m.
to 7:00 a.m. each day, with no specified termination. This means that this
time range will stay in effect until it is manually removed.
To illustrate a more detailed example of the use of the absolute statement, assume an organization is readjusting its network and needs to put into effect portions of an access list from June 1, 2000 at 8:00 p.m. until midnight on July 15, 2000. To configure this time range one would use the following statement:

```
absolute start 20:00 1 June 2000 end 24:00 15 July 2000
```

**RULES**

Similar to other IOS statements, there are certain rules associated with the use of the absolute start and end variables that must be obeyed. One rule, as previously discussed, is the omission of a start or end keyword and associated time and date. A second rule involves the times and dates specified for starting and stopping the time range. As one might logically expect, the end time and end date must be specified as occurring after the start time and date.

To illustrate another example of the use of the absolute statement assume one wants to allow HTTP traffic beginning at 8:00 p.m. on June 1, 2000 but does not wish to specify a termination for the time range. The following use of the absolute statement illustrates how to code a time range that begins at 8:00 p.m. on June 1, 2000 and continues indefinitely.

```
absolute start 20:00 1 June 2000
```

With an appreciation for the use of the absolute statement, one can now focus on the use of the periodic statement.

**THE PERIODIC STATEMENT**

As previously mentioned, unlike the absolute statement one can include multiple periodic statements in a time-based access list. The reason multiple periodic statements are supported is because one may need to specify a noncontiguous period of days and times within an absolute time period. Although one could use a single periodic statement, it is often far easier to use multiple statements for both defining the period and for legibility. This can be visualized by first examining the possible arguments one can enter into the periodic statement.

Exhibit 1 lists the arguments supported by the days-of-the-week variable in the periodic statement. Note that although one could enter six arguments to define all days except Sunday, one could also create two periodic statements: one with the argument “weekdays” and a second with the argument “Saturday.”

To illustrate the use of the periodic statement, assume one wants to restrict Web access to weekends, from Saturday at 7:00 a.m. until Sunday at 7:00 p.m. The following periodic statement would be required:
Note that because the keyword *weekend* denotes Saturday and Sunday, one only has to specify the starting and ending times, and IOS is smart enough to associate the beginning time with Saturday and the ending time with Sunday.

To illustrate another example of the use of periodic statements, assume one wants to specify the time range of 7:00 a.m. to 5:00 p.m. Monday through Friday. From Exhibit 1, one would note that the keyword *weekday* is used to reference the period of Monday through Friday. Thus, the following periodic statement would be used:

```
periodic weekday 07:00 to 17:00
```

As a third example of the use of the periodic statement, assume that one wants the time range to commence at 5:00 p.m. on Saturday evening and terminate at 7:00 a.m. on Monday morning. Note that this time period does not fall into one of the predefined categories of daily, weekday, or weekend. Thus, use two periodic statements could be used to create the required time range. The first periodic statement could use the keyword *weekend* to establish the period from 5:00 p.m. on Saturday through midnight on Sunday; and a second periodic statement would then use Monday as the argument. The two statements are shown below:

```
periodic weekend 07:00 to 19:00
```

```
periodic weekday 07:00 to 17:00
```

```
periodic daily 07:00 to 17:00
```

For a fourth example, consider a slightly more complex situation. Assume one wants the time range to commence at 5:00 p.m. on Saturday evening and terminate at 7:00 a.m. on Monday morning. Note that this time period does not fall into one of the predefined categories of daily, weekday, or weekend. Thus, use two periodic statements could be used to create the required time range. The first periodic statement could use the keyword *weekend* to establish the period from 5:00 p.m. on Saturday through midnight on Sunday; and a second periodic statement would then use Monday as the argument. The two statements are shown below:
As an alternative to the use of two periodic statements, one could use a single statement that has explicit day and time values for the beginning and ending days and times. For example, one could explicitly specify the time period as follows:

```
periodic Saturday 17:00 to Monday 07:00
```

Although one can use a single periodic statement as a substitute for multiple statements, this is only possible when there is a contiguous time period. For example, assume there is a requirement to block access to a Web server every weekday from 11:00 p.m. to midnight and on weekends from 07:00 a.m. on Saturday until midnight on Sunday. In this situation, one would have to use multiple periodic statements.

**USAGE GUIDELINES**

Having an appreciation for the `time-range` command and the absolute and period statements associated with that command, consider now a few command guidelines that are easy to overlook but are very important. First, because the use of a time-range access list is dependent on the router's system clock, it is obviously important to ensure that the clock is appropriately set. A second usage guideline that warrants attention concerns the use of absolute and periodic statements within a `time-range` command. When this situation occurs, it is important to note that periodic statement values are evaluated only after the absolute start time is reached. Similarly, periodic statement values are not further evaluated once the absolute statement end time is reached. To illustrate this important concept, assume one wants to specify a time range of weekends from 8:00 a.m. on Saturday morning until 6:00 p.m. on Sunday evening, from June 15, 2000 until December 31, 2000. To create this time range, first use an absolute statement to define the starting and ending dates. Next, use a periodic statement to define the weekend period previously mentioned. Thus, the applicable `time-range` command and absolute and periodic statements required would be as indicated below:

```
time-range allow-http
   absolute start 08:00 15 June 2000 end 31 December 2000
   periodic weekdays 08:00 to 18:00
```

**APPLICATION EXAMPLE**

Given an understanding of the manner by which we can combine absolute and periodic statements within a time range, let's turn our attention
to the creation of an access list that employs a time range. In doing so, assume an organization has a three port-router, with two Ethernet LAN ports and one serial port. Further assume that the serial port provides a connection to the Internet, while the two Ethernet ports provide a connection to two Ethernet LANs as illustrated in Exhibit 2.

In examining Exhibit 2, assume that one wants to restrict access to the Web server whose IP address is 192.78.46.12 on the 192.78.46.0 network. Further assume that the organization wishes to implement a security policy that blocks Web access from the Internet on weekends from 8:00 a.m. Saturday until 6:00 a.m. Monday. In addition to blocking access from the Internet, also assume that another organizational security policy is to block access to the server from all users on the 205.121.150.0 network between the hours of 8:00 a.m. and noon, Monday through Friday. Last but not least, let’s assume that a manager has requested that the previously mentioned time restrictions be put into effect from July 1 through the end of the year 2000.

Based on the previously mentioned assumptions, an applicable time-based access list would be created. Exhibit 3 lists the statements for the list, which actually represents an access list with time-based features added in IOS Version 12.0. To understand the statements in the access list,
examine the statements as they are listed in Exhibit 3. However, before doing so, a few words concerning the access list are in order.

The requirement to block data from the Internet during a different time than when data packets from the 205.121.150.0 network need to be blocked necessitates two different time ranges, with each range corresponding to the periods one wants to block traffic to the Web server. In creating the access list listed in Exhibit 3, one time range is labeled block-http1, while the second time range is labeled block-http2.

In examining the access list statements contained in Exhibit 3, one must first configure the serial interface of the router as one wants to bar Internet access during one time period. To do so, use the `ip access-group` command to define an access list numbered 101 in the inbound direction.

An examination of the next-to-last line in Exhibit 3 reveals the presence of the `ip access list` statement which blocks the TCP protocol from any host to the Web server whose address is 192.78.46.12. Note that the statement terminates with the term `block-http1`, which represents the name of a previously defined time range. Thus, looking at the top of the access list, one notes that the third line contains the statement `time-range`, which defines the name of the time range as `block-http1`. The following three lines contain one absolute and two periodic statements that define the time range that satisfies the previously denoted assumptions. Thus, the first six statements and the next-to-last statement contained in Exhibit
3 represent the time-based access list that blocks access from the Internet to the Web server for one time range.

Beginning with the statement interface Ethernet 1, that statement and the following four statements, along with the last statement, define a second time range named block-http2 that is applied to the access list number 102 that has one statement. That access list is defined for the Ethernet 1 interface in the inbound direction through the use of the second ip access-group statement. The reason the second access list is applied to the Ethernet 1 interface results from a prior requirement; that is, to block users on the 205.121.150.0 network from accessing the Web server during the previously mentioned time range. Note that the second access-group statement is also applied to the interface in the inbound direction. Also note that because Cisco access lists have a default setting that blocks everything that is not especially allowed in the access list shown in Exhibit 3, this setting in effect blocks everything inbound from the Internet and via the Ethernet 1 interface. Thus, one would obviously add applicable statements to the access lists 101 and 102 to allow any traffic that needs to flow through those interfaces. As indicated, the creation of time-based access lists or, more appropriately, the use of time-based access list features in an access list is easy to implement and provides the ability to control the execution of permit and deny statements based upon predefined time ranges.
SUMMARY

As noted in this article, time-based access lists provide the ability to control the execution of permit and deny access list statements by day and date. Doing so provides the ability to dynamically change security policies by time and add additional flexibility to satisfy organizational requirements.

Gilbert Held is an award-winning author and lecturer. Gil is the author of over 40 books and 250 technical articles covering computer communications. Some of Gil’s recent books include Voice and Data Internetworking, 2nd ed. and Cisco Architecture, both published by McGraw-Hill, and Data Communications Networking, 4th ed., The Complete Modem Reference 3rd ed., and Internetworking LANs and WANs 2nd ed., all published by John Wiley & Sons. He can be reached via e-mail at gil_held@yahoo.com.