Solutions Guide for BIG-IP® Traffic Management Systems

version 9.2.2

MAN-0188-01

Product Version

This manual applies to version 9.2.2 of BIG-IP® Local Traffic Manager™, BIG-IP® Load Balancer Limited™, and BIG-IP® SSL Accelerator™.

Publication Date

This guide was published on December 15, 2005.

Legal Notices

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This class A digital apparatus complies with Canadian I CES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Bal·zs Scheidler <bazsi@balabit.hu>, which is protected under the GNU Public License.

This product includes software developed by Niels M˝ller <nisse@lysator.liu.se>, which is protected under the GNU Public License.

Table of Contents

1

Introducing Solutions for BIG-IP System Traffic Management

- Introducing the BIG-IP system
- Introducing BIG-IP system solutions
- About this guide
- Finding help and technical support resources

Introducing the BIG-IP system

The BIG-IP® system is a port-based, multilayer switch that supports virtual local area network (VLAN) technology. Because hosts within a VLAN can communicate at the data-link layer (Layer 2), a BIG-IP system reduces the need for routers and IP routing on the network. This in turn reduces equipment costs and boosts overall network performance. At the same time, the BIG-IP system's multilayer capabilities enable the system to process traffic at other OSI layers. The BIG-IP system can perform IP routing at Layer 3, as well as manage TCP, UDP, and other application traffic at Layers 4 through 7. The following modules provide comprehensive traffic management and security for many traffic types. The modules are fully integrated to provide efficient solutions to meet any network, traffic management, and security needs.

- **BIG-IP® Local Traffic Manager**
  The BIG-IP system includes local traffic management features that help make the most of network resources. Using the powerful Configuration utility, you can customize the way that the BIG-IP system processes specific types of protocol and application traffic. By using features such as virtual servers, pools, and profiles, you ensure that traffic passing through the BIG-IP system is processed quickly and efficiently, while meeting all of your security needs. For more information, see the ***Configuration Guide for Local Traffic Management***.

- **BIG-IP® Global Traffic Manager**
  The Global Traffic Manager provides intelligent traffic management to your globally available network resources. Through the Global Traffic Manager, you can select from an array of load balancing modes, ensuring that your clients access the most responsive and robust resources at any given time. In addition, the Global Traffic Manager provides extensive monitoring capabilities so the health of any given resource is always available. For more information, see the ***Configuration Guide for Global Traffic Management***.

- **BIG-IP® Link Controller**
  The Link Controller seamlessly monitors availability and performance of multiple WAN connections to intelligently manage bi-directional traffic flows to a site - providing fault tolerant, optimized Internet access regardless of connection type or provider. The Link Controller ensures that traffic is always sent over the best available link to maximize user performance and minimize bandwidth cost to a data center. For more information, see the ***Configuration Guide for the BIG-IP Link Controller***.

- **BIG-IP®Application Security Module**
  The Application Security Module provides web application protection from application-layer attacks. The Application Security Module protects Web applications from both generalized and targeted application layer attacks including buffer overflow, SQL injection, cross-site scripting, and parameter tampering. For more information, see the ***Configuration Guide for the BIG-IP Application Security Module***.

Introducing BIG-IP system solutions

In a typical configuration, the BIG-IP system functions as a device on the network, directing different types of protocol and application traffic to an appropriate destination server. The system accomplishes this by either forwarding the traffic directly to a load balancing server pool, or by sending it to a next-hop router or a pool of routers. The most basic configuration of the BIG-IP system includes two virtual local area networks (VLANs) with one or more BIG-IP interfaces (ports) assigned to each VLAN. Using the BIG-IP system's browser-based Configuration utility, you can implement many configuration scenarios simply by using the default VLAN configuration, and then creating BIG-IP system objects such as a customized virtual server, traffic profile, and load balancing pool.

Getting started

Before you begin configuring a solution, we recommend that you run the Setup utility on the BIG-IP system to configure basic network and network elements such as static and floating self IP addresses, interfaces, and VLANs.

After running the Setup utility, you can use this guide to implement specific configuration scenarios.

Before you begin configuring a solution, we recommend that you complete these tasks:

- Choose a configuration tool.
- Familiarize yourself with additional resources such as other BIG-IP system guides and online help.
- Review the stylistic conventions that appear in this chapter.

Using the Configuration utility

All users need to use the web-based Configuration utility in order to license the system for the first time.

In addition to setting up the management network and initial traffic management software configuration, you use the Configuration utility to perform additional configuration steps necessary for your configuration.

The Configuration utility supports Netscape® Navigator™, version 7.1, or other browsers built on the same engine, such as Mozilla™, Firefox™, and Camino™; and Microsoft® Internet Explorer™ version 6.x and later.

For information on setting user preferences for the Configuration utility, see the ***BIG-IP® Network and System Management Guide***.

About this guide

The chapters contained in this guide provide step-by-step procedures for implementing complete traffic management solutions. For example, Chapter 3, *Basic Web Site and E-Commerce Configuration*, describes how to configure the BIG-IP system objects that you need to set up an array of Web servers that process e-commerce traffic

Additional information

In addition to this guide, there are other sources of the documentation you can use in order to work with the BIG-IP system. The information is organized into the guides and documents described below. The following printed documentation is included with the BIG-IP system.

- **Configuration Worksheet**
  This worksheet provides you with a place to plan the basic configuration for the BIG-IP system.
- **BIG-IP Quick Start Instructions**
  This pamphlet provides you with the basic configuration steps required to get the BIG-IP system up and running in the network.

The following guides are available in PDF format from the ISO image provided with the BIG-IP system. These guides are also available from the first Web page you see when you log in to the administrative web server on the BIG-IP system.

- **Platform Guide**
  This guide includes information about the BIG-IP system. It also contains important environmental warnings.
- **Installation, Licensing, and Upgrades for BIG-IP Systems**
  This guide provides detailed information about installing upgrades to the BIG-IP system. It also provides information about licensing the BIG-IP system software and connecting the system to a management workstation or network.
- **BIG-IP® Network and System Management Guide**
  This guide contains any information you need to configure and maintain the network and system-related components of the BIG-IP system. With this guide, you can perform tasks such as configuring VLANs, assigning self IP addresses, creating administrative user accounts, and managing a redundant system.
- **Configuration Guide for Local Traffic Management**
  This guide contains any information you need for configuring the BIG-IP system to manage local network traffic. With this guide, you can perform tasks such as creating virtual servers and load balancing pools, configuring application and persistence profiles, implementing health monitors, and setting up remote authentication.

Stylistic conventions

To help you easily identify and understand important information, all of our documentation uses the stylistic conventions described here.

Using the solution examples

All examples in this document use only private class IP addresses. When you set up the solutions we describe, you must use valid IP addresses suitable to your own network in place of our sample addresses.

Identifying new terms

To help you identify sections where a term is defined, the term itself is shown in bold italic text. For example, a ***floating IP address*** is an IP address assigned to a VLAN and shared between two computer systems.

Identifying references to products

We refer to all products in the BIG-IP product family as BIG-IP systems. We refer to the software modules by their name; for example, we refer to the Local Traffic Manager module as simply the Local Traffic Manager. If configuration information relates to a specific hardware platform, we note the platform.

Identifying references to objects, names, and commands

We apply bold text to a variety of items to help you easily pick them out of a block of text. These items include web addresses, IP addresses, utility names, and portions of commands, such as variables and keywords. For example, with the **bigpipe self <ip_address> show** command, you can specify a specific self IP address to show by specifying an IP address for the **<ip_address>** variable.

Identifying references to other documents

We use italic text to denote a reference to another document. In references where we provide the name of a book as well as a specific chapter or section in the book, we show the book name in bold, italic text, and the chapter/section name in italic text to help quickly differentiate the two. For example, you can find information about SNMP traps in the ***BIG-IP® Network and System Management Guide***.

Identifying command syntax

We show complete commands in bold Courier text. Note that we do not include the corresponding screen prompt, unless the command is shown in a figure that depicts an entire command line screen. For example, the following command shows the configuration of the specified pool name:

```
bigpipe self <ip_address> show
```

or

```
b self <ip_Address> show
```

Table 1.1 explains additional special conventions used in command line syntax.

Item in text	Description
\	Indicates that the command continues on the following line, and that users should type the entire command without typing a line break.
< >	Identifies a user-defined parameter. For example, if the command has **<your name>**, type in your name, but do not include the brackets.
|	Separates parts of a command.
[]	Indicates that syntax inside the brackets is optional.
...	Indicates that you can type a series of items.

Table 1.1 *Command line syntax conventions*

Finding help and technical support resources

You can find additional technical documentation and product information in the following locations:

- **Online help for local traffic management**
  The Configuration utility has online help for each screen. The online help contains descriptions of each control and setting on the screen. Click the Help tab in the left navigation pane to view the online help for a screen.
- **Welcome screen in the Configuration utility**
  The Welcome screen in the Configuration utility contains links to many useful web sites and resources, including:
  - The F5 Networks Technical Support web site
  - The F5 Solution Center
  - The F5 DevCentral web site
  - Plug-ins, SNMP MIBs, and SSH clients
- **F5 Networks Technical Support web site**
  The F5 Networks Technical Support web site, **http://tech.f5.com**, provides the latest documentation for the product, including:
  - Release notes for the BIG-IP system, current and past
  - Updates for guides (in PDF form)
  - Technical notes
  - Answers to frequently asked questions
  - The Ask F5 natural language question and answer engine.

  To access this site, you need to register at **http://tech.f5.com**.

2

Configuring nPath Routing

- Introducing nPath routing
- Configuring nPath routing
- Setting timers for nPath configurations

Introducing nPath routing

With the nPath routing configuration, you can route outgoing server traffic around the BIG-IP system directly to an outbound router. This method of traffic management increases outbound throughput because packets do not need to be transmitted to the BIG-IP system for translation and forwarding to the next hop. Figure 2.1 shows an nPath configuration.

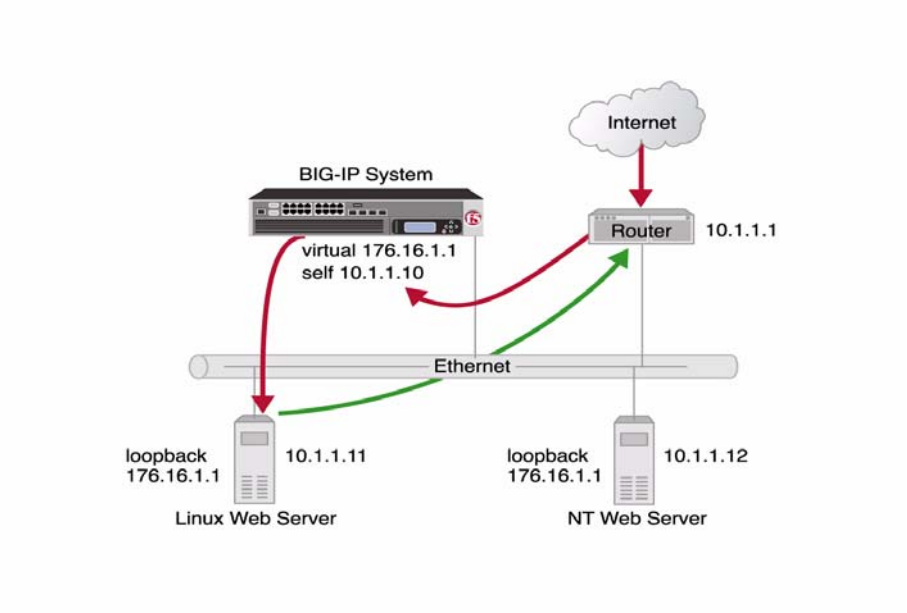

Figure 2.1 *An example of nPath configuration*

◆ Note

The type of virtual server that processes the incoming traffic must be a transparent, non-translating type of virtual server .

In bypassing the BIG-IP system on the return path, nPath routing departs significantly from a typical load-balancing configuration. In a typical load-balancing configuration, the destination address of the incoming packet is translated from that of the virtual server to that of the server being load balanced to, which then becomes the source address of the returning packet. A default route set to the BIG-IP system then sees to it that packets returning to the originating client return through the BIG-IP system, which translates

the source address back to that of the virtual server. The nPath configuration differs from the typical load-balancing configuration, as you can see in the following section.

◆ Note

Do not attempt to use nPath routing for Layer 7 traffic. Certain traffic features do not work properly if Layer 7 traffic bypasses the BIG-IP system on the return path. An example of such a feature is HTTP response compression.

Configuring nPath routing

The nPath routing configuration differs from the typical BIG-IP load balancing configuration in the following ways:

- The default route on the content servers must be set to the router's internal address (**10.1.1.1** in Figure 2.1) rather than to the BIG-IP system's floating self-IP address (**10.1.1.10**). This causes the return packet to bypass the BIG-IP system.
- If you plan to use an nPath configuration for TCP traffic, you must create a Fast L4 profile with the following custom settings:
  - Enable the **Loose Close** setting. When you enable the **Loose Close** setting, the TCP protocol flow expires more quickly, once a TCP FIN packet is seen. (A ***FIN packet*** indicates the tearing down of a previous connection.)
  - Set the **TCP Close Timeout** setting to the same value as the profile idle timeout if you expect half closes. If not, you can set this value to 5 seconds.
- Because address translation and port translation have been turned off, the incoming packet arrives at the pool member it is load balanced to with the virtual server address (**176.16.1.1** in Figure 2.1), not the address of the server. For the server to respond to that address, that address must be configured on the loopback interface of the server and configured for use with the server software.

You need to complete the following tasks to configure the BIG-IP system to use nPath routing:

- Create a custom Fast L4 profile.
- Create a pool that contains the content servers.
- Define a virtual server with port and address translation disabled and assign the custom Fast L4 profile to it.
- Configure the virtual server address on each server loopback interface.
- Set the default route on your servers to the router's internal IP address.

◆ Note

Before you begin these tasks, log in to the Configuration utility.

Creating a custom Fast L4 profile

The first task you must complete to create an nPath routing configuration is to create a custom Fast L4 profile.

To create a custom Fast L4 profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   This displays the HTTP Profiles screen.
2. From the Protocol menu, choose Fast L4.
   The Fast L4 Profiles screen opens.
3. To create a custom profile, click **Create**.
   The New Fast L4 Profile screen opens.
4. In the New Fast L4 Profile screen, set the following attributes.
   a) In the **Name** box, type a name for the profile.
   b) Enable the loose close option by checking the corresponding **Select** box on the right side of the screen, and then checking the **Loose Close** box.
   c) Set the **TCP Idle Timeout** setting according to the type of traffic the virtual server is going to handle. For additional information about setting this timeout, see *Setting timers for nPath configurations*, on page 2-7.
5. Click **Finished**.

Creating a server pool for nPath routing

After you create a custom Fast L4 profile, you need to create a server pool.

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. To create a new pool, click **Create**.
   The New Pool screen opens.
3. Type a pool name and add the member addresses for each of the servers. (For additional information about configuring a pool, click the **Help** tab.)
4. Click **Finished**.

Configuration note

For this example, you create an HTTP pool named ***http_pool*** *containing the following members:*
10.1.1.11
10.1.1.12

Creating a virtual server

After you create a server pool, you need to create a virtual server that references the customer Fast L4 profile and pool you created in the last two tasks.

To create a standard virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. To create a new virtual server, click **Create**.
   The New Virtual Server screen opens.
3. Type the virtual server name, select a destination type, and type the IP address.
4. For the **Type** setting, select **Performance (Layer 4)**.
5. Set the following attributes:
   a) For **Protocol**, select either **UDP**, **TCP**, or ***All Protocols** from the list.
   b) For **Protocol Profile (Client)**, select the name of the custom Fast L4 profile that you created.
   c) Clear the **Address Translation** check box to disable address translation.
   d) Clear the **Port Translation** check box to disable port translation.
   e) In the Resources section, choose the pool you created that contains the content servers.
6. Click **Finished**.

Configuration notes

For this example, you create a virtual server ***176.16.1.1*** *that references the HTTP pool named* ***http_pool****.*

Configuring the virtual server on the content server loopback interface

You must place the IP address of the virtual server (**176.16.1.1** in Figure 2.1 on page 2-1) on the loopback interface of each server. Most UNIX variants have a loopback interface named **lo0**. Microsoft® Windows® has an MS Loopback interface in its list of network adaptors. For some versions of Windows, you must install the loopback interface using the installation CD. Consult your server operating system documentation for information about configuring an IP address on the loopback interface. The loopback interface is ideal for the nPath configuration because it does not participate in the ARP protocol.

Setting the route for inbound traffic

For inbound traffic, you must define a route through the BIG-IP system self IP address to the virtual server. In the example, this route is **176.16.1.1**, with the external self address **10.1.1.10** as the gateway.

◆ Note

You need to set this route only if the virtual server is on a different subnet than the router.

For information about how to define this route, please refer to the documentation provided with your router.

Setting timers for nPath configurations

When you create an nPath configuration, the BIG-IP system sees only client requests. Therefore, the timer for the connection timeout is only reset when clients transmit. In general, this means the timeout for an nPath connection should be at least twice as long as for a comparable connection where BIG-IP system sees both client requests and node responses. Following are descriptions of scenarios for setting the timers for UDP and TCP traffic.

Guidelines for configuring timeouts for UDP traffic

When you configure nPath for UDP traffic, the BIG-IP system tracks packets sent between the same source and destination address to the same destination port as a connection. This is necessary to ensure that client requests that are part of a session always go to the same server. Therefore, a UDP connection is really a form of persistence, since UDP is a connectionless protocol. To calculate the timeout for UDP, estimate the maximum amount of time that a server transmits UDP packets before a packet is sent by the client. In some cases, the server might transmit hundreds of packets over several minutes before ending the session or waiting for a client response.

Guidelines for configuring timeouts for TCP traffic

When you configure nPath for TCP traffic, the BIG-IP system sees only the client side of the connection. For example, in the TCP three-way handshake, the BIG-IP system sees the SYN from the client to the server, and does not see the SYN acknowledgement from the server to the client, and does see the acknowledgement of the acknowledgement from the client to the server. The timeout for the connection should match the combined TCP retransmission timeout (RTO) of the client and the node as closely as possible to ensure that all connections are successful. The maximum initial RTO observed on most UNIX and Windows systems is approximately 25 seconds. Therefore, a timeout of 51 seconds should adequately cover the worst case. Once a TCP session is established, an adaptive timeout is used. In most cases, this results in a faster timeout on the client and node. Only if your clients are on slow, lossy networks should you ever need a higher TCP timeout for established connections. Once a FIN packet is received from the client, the **TCP Close Timeout** option is used to more aggressively remove connections from the BIG-IP system.

3

Basic Web Site and E-Commerce Configuration

- Working with a basic web site and e-commerce configuration
- Configuring a basic e-commerce site

Working with a basic web site and e-commerce configuration

The most common use for the BIG-IP system is distributing traffic across an array of web servers that host standard web traffic, including e-commerce traffic. Figure 3.1 shows a configuration where a BIG-IP system load balances two sites: **www.MySite.com** and **store.MySite.com**. The **www.MySite.com** site provides standard web content, and the **store.MySite.com** site is the e-commerce site that sells items to **www.MySite.com** customers.

To set up load balancing for these sites, you need to create two pools that are referenced by two virtual servers, one for each site. Even though the sites are related and they may even share the same IP address, each requires its own virtual server because it uses a different port to support its particular protocol: port **80** for the HTTP traffic going to **www.MySite.com**, and port **443** for the SSL traffic going to **store.MySite.com**. Note that this is true even when there is a port **80** and port **443** on the same physical server, as in the case of **Server2**.

Note

All examples in this document use only private class IP addresses. When you set up the solutions we describe, you must use valid IP addresses suitable to your own network in place of our sample addresses.

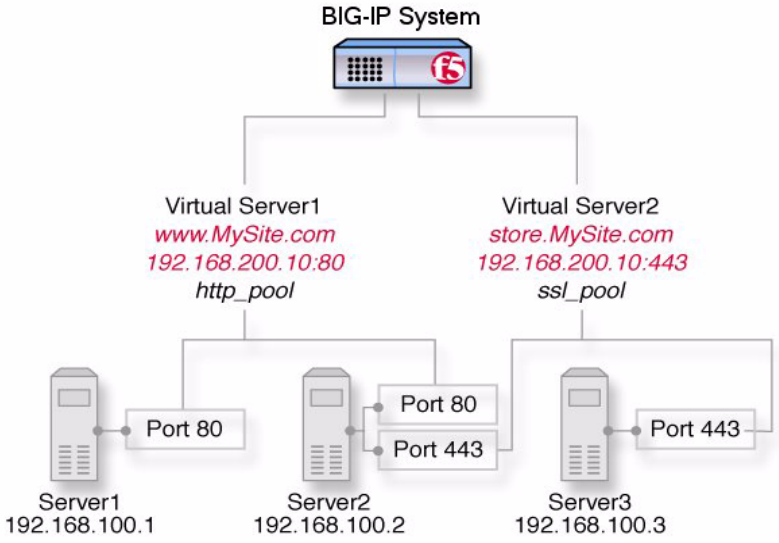

Figure 3.1 *A basic load balancing configuration*

Configuring a basic e-commerce site

To configure the e-commerce site, you need to complete the following tasks in order:

- Create the load balancing pools
- Create virtual servers for the inbound traffic

Creating load balancing pools

The first task in a basic configuration is to define the two load balancing pools: a pool to load balance HTTP connections, and a pool to load balance SSL connections. As shown in Figure 3.1, the two servers for the HTTP pool are **192.168.100.1:80** and **192.168.100.2:80** (**Server1** and **Server 2**). The two servers for the SSL pool are **192.168.100.2:443** and **192.168.100.3:443** (**Server2** and **Server 3**).

Use the Configuration utility to create these two pools. For additional information about configuring a pool, see the online help.

To create a pool for load balancing HTTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool.
   In the example in Figure 3.1, on page 3-1, this pool name is **http_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   In the example in Figure 3.1, on page 3-1, these pool members are **192.168.100.1:80** and **192.168.100.2:80**.
5. Click **Finished**.

To create a pool for load balancing SSL traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool.
   In the example in Figure 3.1, on page 3-1, this pool name is **ssl_pool**.

4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   In the example in Figure 3.1, on page 3-1, these pool members are **192.168.100.2:443** and **192.168.100.3:443**.
5. Click **Finished.**

Creating virtual servers

The next task in a basic configuration is to define the virtual servers that reference the HTTP and SSL pools, respectively. You use the Configuration utility to create these virtual servers. For additional information about configuring a virtual server, click the **Help** button.

To define a virtual server for HTTP traffic

1. On the Main tab of the navigation pane expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_http**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server, such as **192.168.200.10:80**.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, locate the **HTTP Profile** setting and select **http**.
7. In the Resources area of the screen, locate the **Default Pool** setting and select **http_pool**.
8. Click **Finished**.

To define a virtual server for SSL traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_ssl**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server, such as **192.168.200.10:443**.
5. In the **Service Port** box, type **443**, or select **HTTPS** from the list.

6. In the Configuration area of the screen, locate the **SSL Profile (Client)** setting and select **clientssl**.
7. In the Resources area of the screen, locate the **Default Pool** setting and select **ssl_pool**.
8. Click **Finished**.

4

Installing a BIG-IP System without Changing the IP Network

- Installing a BIG-IP system without changing IP networks
- Configuring the BIG-IP system for the same IP network

Installing a BIG-IP system without changing IP networks

A combination of several features of the BIG-IP system allows you to place a BIG-IP system in a network without changing the existing IP network.

Figure 4.1 shows the data center topology before you add the BIG-IP system. The data center has one LAN, with one IP network, **10.0.0.0**. The data center has one router to the Internet, two web servers, and a back-end mail server.

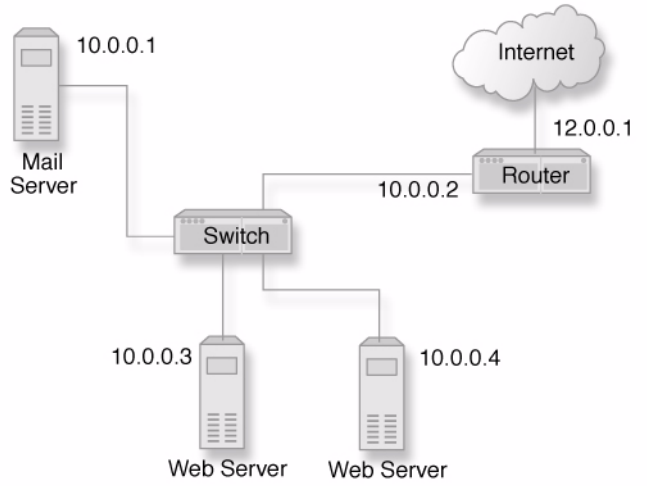

Figure 4.1 *Existing data center network structure*

The existing data center structure does not support load balancing or high availability. Figure 4.2 is an example of the data center topology after you add the BIG-IP system.

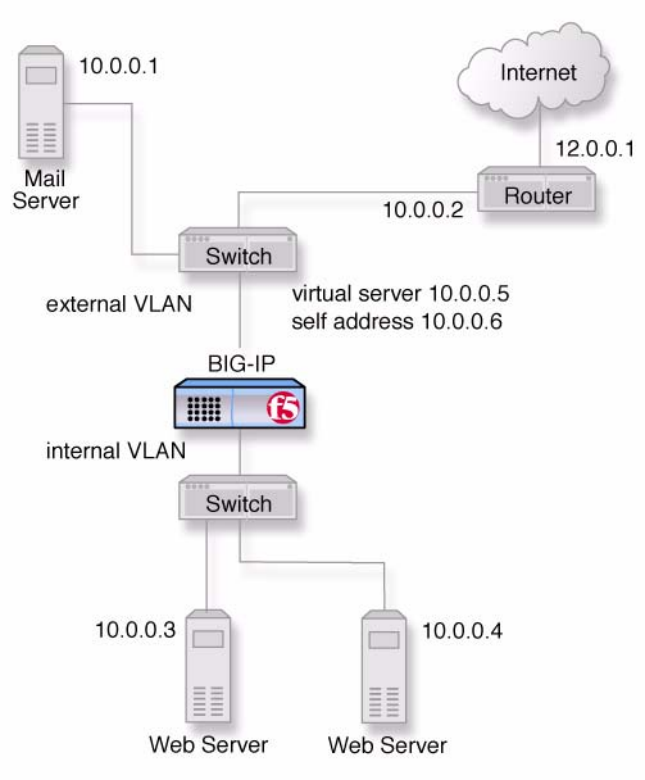

Figure 4.2 *New structure after adding the BIG-IP system*

Both the internal and external interfaces of the BIG-IP system are on the same IP network, **10.0.0.0**, but they are effectively on different LANs.

Note that a second switch has been introduced in Figure 4.2. This switch would be eliminated in a configuration using a BIG-IP system.

Configuring the BIG-IP system for the same IP network

To configure the BIG-IP system for this solution, you must create a VLAN group, a pool of web servers, and a virtual server: More specifically, you must:

- **Remove the self IP addresses from the individual VLANs**
  Routing is handled by the self IP address you create for the VLAN group.
- **Create a VLAN group**
  Create a VLAN group that includes the internal and external VLANs. This enables layer 2 forwarding. (Layer 2 forwarding causes the two VLANs to behave as a single network.)
- **Create a self IP for the VLAN group**
  The self IP for the VLAN group provides a route for packets destined for the network.
- **Create a pool of web servers**
  Create a pool that contains the web servers that you want to load balance.
- **Create a virtual server**
  Create a virtual server that load balances the web servers.

Note

*This example assumes that you are using the default **internal** and **external** VLAN configuration with self IP addresses on each of the VLANs that are on the same IP network on which you are installing the BIG-IP system.*

Important

*The default route on each content server should be set to the IP address of the router. In this example, you set the default route to **10.0.0.2**.*

Removing the self IP addresses from the individual VLANs

Remove the self IP addresses from the individual VLANs. After you create the VLAN group, you will create another self IP address for the VLAN group for routing purposes. The individual VLANs no longer need their own self IP addresses.

WARNING

We recommend that you perform this step from the console or from a self IP address you are not going to delete. If you are connected from a remote workstation though a self IP address you are going to delete, you will be disconnected when you delete it.

To remove the self IP addresses from the default VLANs

1. On the Main tab of the navigation pane, expand **Network**, and click **Self IPs**.
   The Self IPs screen opens.
2. Using the IP Address and VLANs columns, locate the self IP addresses for the Internal and External VLANs.
3. To the left of each self IP address you want to delete, check the Select box.
4. Click **Delete**.
   A confirmation screen appears.
5. Click **Delete** again.

Creating a VLAN group

Create a VLAN group that includes the internal and external VLANs. Packets received by a VLAN in the VLAN group are copied onto the other VLAN in the group. This allows traffic to pass through the BIG-IP system on the same IP network.

To create a VLAN group

1. On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLANs screen opens.
2. From the VLAN Groups menu, choose List.
   This opens the VLAN Groups screen.
3. In the upper-right corner of the screen, click **Create**.
   This opens the New VLAN Group screen.
4. In the **Name** box, type the name **myvlangroup**.
5. For the **VLANs** setting, from the **Available** box select the **internal** and **external** VLAN names, and click the Move button (**<<**) to move the VLAN names to the **Members** box.
6. Click **Finished**.

Creating a self IP address for the VLAN group

The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this example, the path to the IP network **10.0.0.0** is actually through more than one VLAN. Since IP routers are designed to have only one physical route to a network, a routing conflict can occur. The self IP address feature on the BIG-IP system allows you to resolve the routing conflict by putting a self IP address on the VLAN group.

To create a self IP address for a VLAN group

1. On the Main tab of the navigation pane, expand **Network,** and click **Self IPs**.
   The Self IPs screen opens.
2. In the upper-right corner of the screen, click **Create**.
3. In the **IP Address** box, type a self IP address for the VLAN group.
   In the example shown in Figure 4.2, on page 4-2, this IP address is **10.0.0.6**.
4. In the **Netmask** box, type a netmask for the self IP address.
5. For the **VLAN** setting, select the name **myvlangroup** from the list.
6. Click **Finished**.

Creating a pool of web servers

After you create the network environment for the BIG-IP system, you can create the pool of web servers you want to load balance.

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **myweb_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   In our example, pool members are **10.0.0.3:80** and **10.0.0.4:80**.
5. Click **Finished**.

Creating a virtual server

After you create the pool of web servers you want to load balance, you can create the virtual server.

To create a virtual server

1. On the Main tab, of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.

3. In the **Name** box, type a name for the virtual server, such as **vs_myweb**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address.
   Continuing with our example, this address would be **10.0.0.5**.
5. From the **Service Port** list, select ***All Ports**.
6. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the pool you created using the previous procedure.
   In our example, this pool name is **myweb_pool**.
7. Click **Finished**.

5

Web Hosting for Multiple Customers

- Introducing multiple customer hosting
- Hosting multiple customers using an external switch
- Directly hosting multiple customers

Introducing multiple customer hosting

You can use the BIG-IP system to load balance and provide hosting services for multiple customers.

In this example, the BIG-IP system has an interface (5.1) assigned to three VLANs on a network. The three VLANs are **vlanA**, **vlanB**, and **vlanC**. Interface 5.1 processes traffic for all three VLANs. Note that each VLAN contains two servers, and serves a specific customer.

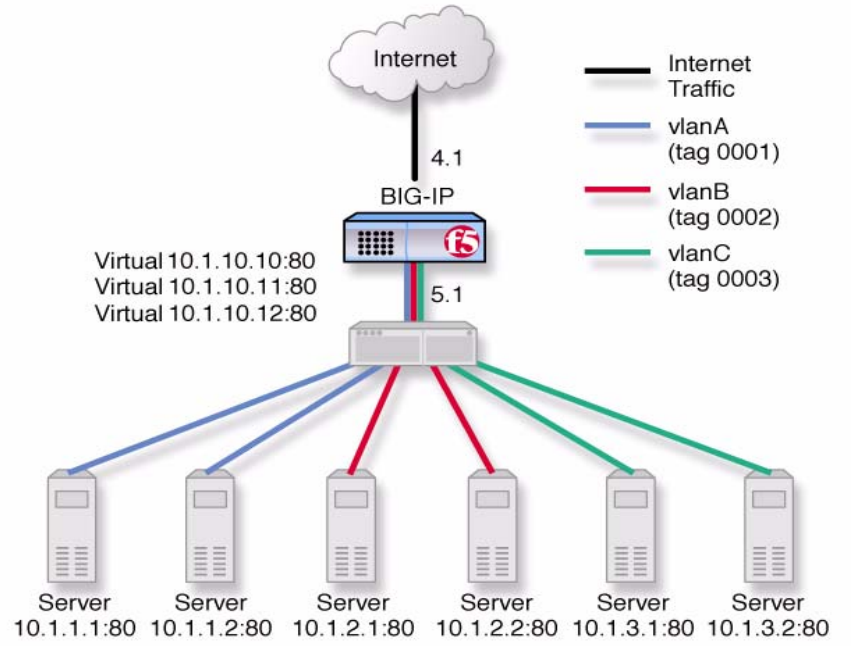

Figure 5.1 *An example of multiple site hosting*

Hosting multiple customers using an external switch

To configure the BIG-IP system for this solution, you must complete the following tasks:

- Create VLANs with tagged interfaces.
- Create pools of web servers to which you want to load balance traffic.
- Create a virtual server that load balances traffic to the web servers.

Creating VLANs with tagged interfaces

The first task in configuring the BIG-IP system for multiple-customer hosting is creating VLANs with tagged interfaces. In this procedure, you assign the same interface to each VLAN that you create, and you assign the interface as a tagged interface. (For more information on tagged interfaces, see the ***BIG-IP® Network and System Management Guide***.)

For example in Figure 5.1, on page 5-1, there are three VLANs, where each VLAN processes traffic for a different subnet. Thus, **vlanA** processes traffic for the **10.1.1** subnet, **vlanB**, processes traffic for the **10.1.2** subnet, and **vlanC** processes traffic for the **10.1.3** subnet. The interface assigned to all three VLANs is 5.1.

To create a VLAN with a tagged interface

1. On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLAN screen opens.
2. In the upper-right corner of the screen, click **Create**.
   This displays the settings to configure for the VLAN.
3. Enter the VLAN name and tag number.
   If you do not provide a tag number, the BIG-IP system automatically generates a number. In Figure 5.1, on page 5-1, an example of a VLAN name and tag number is **vlanA**, with a tag number of **0001**.
4. For the **Interfaces** setting, from the **Available** box select the name of an interface on your internal network, and click the Move button (**<<**) to move the interface name to the **Tagged** box.
   This assigns the selected interface to the VLAN, as a tagged interface. In our example, the interface is 5.1.
5. Click **Finished**.

Creating load balancing pools

After you create the VLANs for the BIG-IP system, create three load balancing pools, one for each VLAN. In Figure 5.1, on page 5-1, each pool has two pool members.

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **customerA_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   For example, in Figure 5.1, on page 5-1, the pool members for **vlanA** are **10.1.1.1:80** and **10.1.1.2:80**. The pool members for **vlanB** are **10.1.2.1:80** and **10.1.2.2:80**, and the pool members for **vlanC** are **10.1.3.1:80** and **10.1.3.2:80**.
5. Click **Finished**.

Creating virtual servers

After you create the web server pools that you want to load balance, create a virtual server for each pool.

To create a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_customerA**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server, such as **10.1.10.10:80**.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, locate the **HTTP Profile** setting and select **http**.

7. In the Resources area of the screen, locate the **Default Pool** setting and select the pool corresponding to the virtual server you are creating.
   For example, for **vs_customerA**, you would select the pool **customerA_pool**. For **vs_customerB**, you would select the pool **customerB_pool**, and so on.
8. Click **Finished**.

Directly hosting multiple customers

The configuration shown in Figure 5.1, on page 5-1, uses an external switch between the BIG-IP system and the server nodes. However, another way to implement this solution is to remove the external switch, and instead use multiple interfaces on the BIG-IP system to directly host traffic for multiple customers. With this scenario, it is still necessary to configure the VLANs, but you can configure them with untagged instead of tagged interfaces. Figure 5.2 shows an example of this scenario.

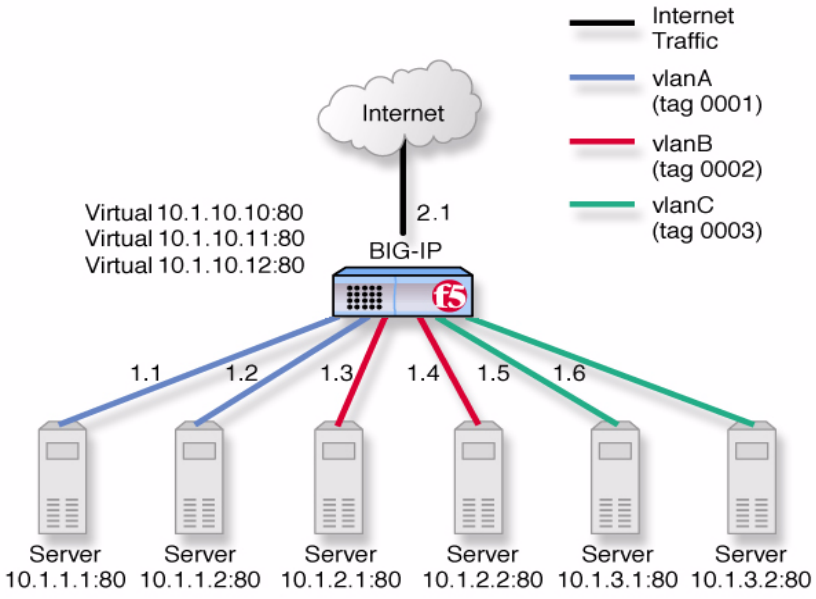

Figure 5.2 *Multiple customer hosting using VLAN switching*

In Figure 5.2, two BIG-IP system interfaces are assigned to each VLAN. For example, interfaces 1.1 and 1.2 are assigned to the **vlanA** VLAN. Each interface is assigned to a VLAN as an untagged interface.

The first scenario, shown in Figure 5.1, on page 5-1, requires an additional switch, but requires the use of only one interface on the internal network. The second scenario, shown in Figure 5.2, on page 5-4, removes the need for an additional switch, but requires the use of multiple BIG-IP system interfaces.

Creating VLANs with untagged interfaces

The first task in configuring the BIG-IP system for directly hosting multiple customers is to create VLANs, adding untagged interfaces to them. (For more information on tagged interfaces, see the ***BIG-IP® Network and System Management Guide***.)

To create VLANs with untagged interfaces

1. On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLAN screen opens.
2. In the upper-right corner of the screen, click **Create**.
   This displays the settings to configure for the VLAN.
3. Enter the VLAN name and tag number.
   If you do not provide a tag number, the BIG-IP system automatically generates a number. In Figure 5.2, on page 5-4, an example of a VLAN name and tag number is **vlanA**, with a tag number of **0001**.
4. For the **Interfaces** setting, from the **Available** box select the name of an interface on your internal network, and click the Move button (**>>**) to move the interface name to the **Untagged** box.
   This assigns the selected interface to the VLAN, as an untagged interface. In Figure 5.2, on page 5-4, **vlanA** as interfaces 1.1 and 1.2 assigned to it. **vlanB** has interfaces 1.3 and 1.4 assigned to it, and **vlanC** has interfaces 1.5 and 1.6 assigned to it.
5. Click **Finished**.

Once you have created your VLANs and assigned untagged interfaces to them, you can create the pools and virtual servers, just as you did in the section *Hosting multiple customers using an external switch*, on page 5-2.

6

A Simple Intranet Configuration

- Working with a simple intranet configuration
- Creating the simple intranet configuration

Working with a simple intranet configuration

The simple intranet solution described in this chapter is commonly found in a corporate intranet (see Figure 6.1). In this scenario, the BIG-IP system performs load balancing for several different types of connection requests:

- HTTP connections to the company's intranet web site. The BIG-IP system load balances the two web servers that host the corporate intranet web site, **Corporate.main.net**.
- HTTP connections to Internet content. These are handled through a pair of cache servers that are also load balanced by the BIG-IP system.
- Non-HTTP connections to the Internet.

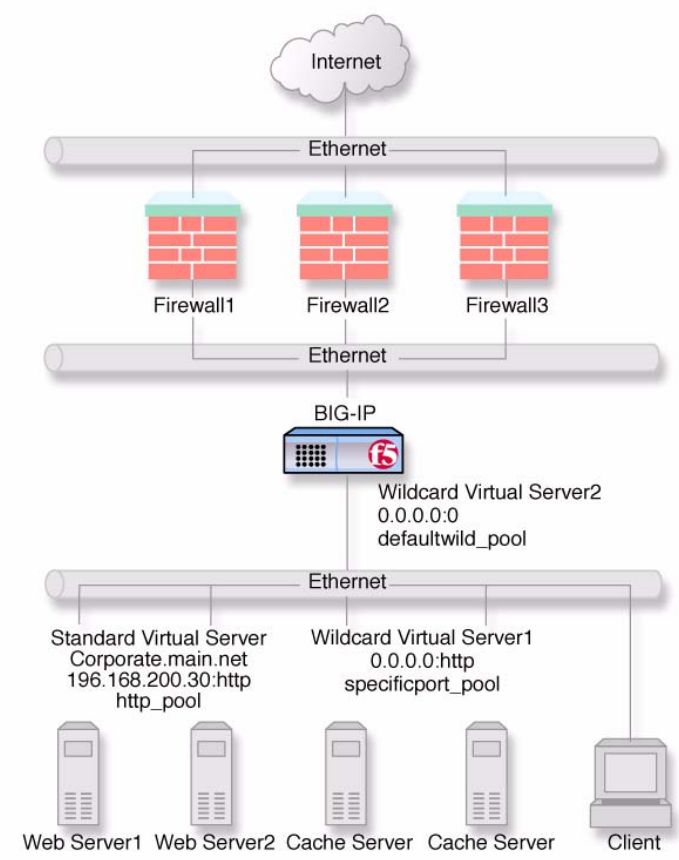

Figure 6.1 *A simple intranet configuration*

As Figure 6.1 shows, the non-intranet connections are handled by wildcard virtual servers, that is, servers with the IP address **0.0.0.0**. The wildcard virtual server that is handling traffic to the cache servers is port specific, specifying port **80** for HTTP requests. This way all HTTP requests not matching an IP address on the intranet are directed to the cache server. The wildcard virtual server handling non-HTTP requests is a ***default*** wildcard server. A default wildcard virtual server is one that uses only port **0**. This makes it a catch-all match for outgoing traffic that does not match any standard virtual server or any port-specific wildcard virtual server.

Creating the simple intranet configuration

To create this configuration, you need to complete the following tasks in order:

- **Create load balancing pools**
  Create pools for the intranet servers you want to load balance and one for the cache server.
- **Create virtual servers**
  Create the virtual servers for each pool and for the non-HTTP requests.

Creating pools

The first task in a basic configuration is to define the two load balancing pools: a pool for the intranet content servers and a pool for the Internet cache servers.

To create pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **http_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   For example, in Figure 6.1, on page 6-1, the pool members for http_pool are **192.168.100.10:80** and **192.168.100.11:80**. The pool members for **specificport_pool** are **192.168.100.20:80** and **192.168.100.21:80**.
5. Click **Finished**.

Creating virtual servers

The next task in a basic configuration is to create the virtual servers that reference **http_pool** and **specificport_pool**, respectively. You must also create a forwarding virtual server (with no pool) for remaining Internet traffic.

To create a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.

3. In the **Name** box, type a name for the virtual server, such as **vs_http**, **vs_specificport**, or **vs_non-http**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server. For example, you can assign the IP address **192.168.200.30:80** to the virtual server that processes HTTP traffic. For load balancing connections to cache servers, you can assign the address **0.0.0.0:80** to the virtual server, making it a wildcard virtual server. To create a forwarding virtual server, you can assign the address **0.0.0.0:0**.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, locate the **Type** setting and do the following:
   a) Select **Standard** if the virtual server is to process HTTP traffic to an intranet web site or to cache servers.
   b) Select **Forwarding (IP)** if the virtual server is to forward outgoing non-HTTP traffic.
7. If you are creating a virtual server to process HTTP connections to an intranet web site, locate the **HTTP Profile** setting and select **http**.
8. In the Resources area of the screen, locate the **Default Pool** setting and select the pool corresponding to the virtual server you are creating.
   For example, for **vs_http**, you would select the pool **http_pool**.
   Note: *If you are creating a Forwarding (IP) virtual server, you do not select a pool.*
9. Click **Finished**.

7

Load Balancing ISPs

- Introducing ISP load balancing
- Configuring ISP load balancing
- Configuring address translation for outbound traffic

Introducing ISP load balancing

You may find that as your network grows, or network traffic increases, you need to add an additional connection to the internet. You can use this configuration to add an additional Internet connection to your existing network. Figure 7.1 shows a network configured with two Internet connections.

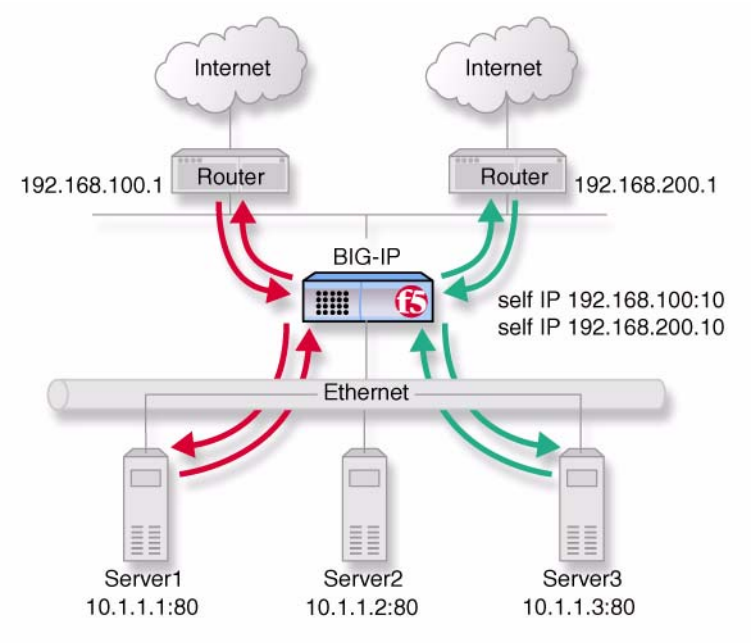

Figure 7.1 *An example of an additional internet connection*

This type of configuration requires you to configure network address translation (NAT) on your routers. If your routers cannot perform NAT, you can use the VLAN SNAT automap feature on the BIG-IP system.

Configuring ISP load balancing

When you set up ISP load balancing, you have several tasks to complete on the BIG-IP system:

- **Create two load balancing pools**
  Define one pool that load balances the content servers. The other pool balances the inside addresses of the routers.
- **Configure virtual servers for inbound and outbound traffic**
  Configure virtual servers to load balance inbound connections across the servers, and one to load balance outbound connections across the routers.
- **Configure NATs or a SNAT automap for outbound traffic**
  Configure NATs or SNAT automap for outbound traffic so that replies arrive though the same ISP the request went out on.

Creating pools for an additional Internet connection

First, create one pool that load balances the content servers, and one pool to load balance the routers.

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **content_pool** or **router_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   For example, in Figure 7.1, on page 7-1, the pool members for pool **content_pool** are **10.1.1.1:80**, **10.1.1.2:80**, and **10.1.1.3:80**. The pool members for pool **router_pool** are **192.168.100.1:0** and **192.168.200.1:0**.
5. Click **Finished**.

Creating virtual servers for an additional Internet connection

After you create the pools, you can configure the two virtual servers, one to load balance inbound connections to the servers, and one to load balance outbound connections to the routers.

To create a virtual server for inbound content server traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_content**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
   For example, you could assign the IP address **172.100.12.20:80**.
5. For the **Service Port** setting, type a port number, or select a service from the list.
6. If the traffic to be load balanced is of a certain type, select the profile type that matches the connection type.
   For example, if the traffic to be load balanced is HTTP traffic, locate the **HTTP Profile** setting and select **http**.
7. In the Resources area of the screen, locate the **Default Pool** setting and select the pool corresponding to the virtual server you are creating.
   For example, for **vs_content**, you would select the pool **content_pool**.
8. Click **Finished**.

To create a virtual server for outbound traffic for routers

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_routers**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
   For example, you can assign the IP address **0.0.0.0:0** to the virtual server, making it a wildcard virtual server.
5. In the Resources area of the screen, locate the **Default Pool** setting and select the pool corresponding to the virtual server you are creating.
   For example, for **vs_routers**, you would select the pool **router_pool**.
6. Click **Finished**.

Configuring address translation for outbound traffic

You must now set up address translation for outbound traffic so that replies arrive through the same ISP that the request initially came through. Specifically, you must either configure your routers so that they perform network address translation (NAT), or you must configure SNAT automapping. You must also assign self IP addresses to the external VLAN.

Note

For instructions on configuring routers to perform network address translation, refer to your router documentation.

To configure address translation for outbound traffic, you must:

- Assign IP-specific self IP addresses to the BIG-IP system external VLAN, corresponding to the IP networks of the two routers.
- Enable SNAT automap for each of the external VLAN self IP addresses and the internal VLAN.

To create self IP addresses for the external VLAN

1. On the Main tab of the navigation pane, expand **Network**, and click **Self IPs.**
   The Self IP screen opens.
2. In the upper-right corner of the screen, click **Create**.
   This displays the settings that you can configure for a self IP address.
3. In the **IP Address** box, type a self IP address that matches the network of the router.

   Note: *Verify that the inside IP network address of the router is enabled.*
4. From the **VLAN** list, select **external**.
5. Click **Repeat**.
6. Create another self IP address for the external VLAN.
7. Click **Finished**.

To enable SNAT automap for internal and external VLANs

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SNATs.**
   The SNATs screen opens.
2. In the upper-right corner, click **Create**.
   The New SNAT screen opens.
3. In the **Name** box, type a unique name for the SNAT.
4. From the **Translation** list, select **Automap.**

5. From the **VLAN Traffic** list, select **Enabled On**.
   This displays the **VLAN List** setting.
6. For the **VLAN List** setting, from the **Available** box select the **internal** and **external** VLAN names, and click the Move button (**<<**) to move the VLAN names to the **Selected** box.
7. Click **Finished.**

8

Load Balancing HTTP Traffic with Source Address Affinity Persistence

- Introducing basic HTTP load balancing
- Configuring HTTP load balancing with source address affinity persistence

Introducing basic HTTP load balancing

Many computing environments want to use a BIG-IP system to intelligently manage their HTTP traffic. You can easily control your HTTP traffic by implementing a BIG-IP system feature known as an HTTP profile. An HTTP profile is a group of settings that affect the behavior of HTTP traffic. An HTTP profile defines the way that you want the BIG-IP system to manage HTTP traffic.

You can use the default HTTP profile, with all of its default values, or you can create a custom HTTP profile. When you create a custom HTTP profile, you not only modify the setting values, but you can enable more advanced features such as data compression of server responses.

When you configure the BIG-IP system to manage HTTP traffic, you can also implement simple session persistence, also known as source address affinity persistence. ***Source address affinity persistence*** directs session requests to the same server based solely on the source IP address of a packet. To implement source address affinity persistence, the BIG-IP system offers a default persistence profile that you can implement. Just as for HTTP, you can use the default profile, or you can create a custom simple persistence profile.

The remainder of this chapter describes how to set up a basic HTTP load balancing scenario and source address affinity persistence, using the default HTTP and persistence profiles. For detailed information on managing HTTP traffic and setting up source address affinity persistence, see the ***Configuration Guide for Local Traffic Management***.

Configuring HTTP load balancing with source address affinity persistence

To set up basic HTTP load balancing with persistence that is based on source IP addresses, you need to create a load balancing pool, and then create a virtual server to process the HTTP traffic and send it to the pool. Because this solution configures HTTP load balancing and session persistence using the default HTTP and source address affinity profiles, you do not need to specifically configure these profiles. Instead, you simply configure some settings on the virtual server when you create it.

Creating a pool

The first task in a basic configuration is to create a load balancing pool to load balance HTTP connections. Use the Configuration utility to create this pool. For more detailed information about configuring a pool, see the ***Configuration Guide for Local Traffic Management***.

To create a pool for load balancing HTTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **http_pool**.
4. For the **Health Monitors** setting, from the **Available** box select **http**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) In the **Service Port** box, type **80**, or select **HTTP**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
6. Click **Finished**.

Creating a virtual server

The next task in a basic configuration is to define a virtual server that references the HTTP pool. You use the Configuration utility to create the virtual server. For more information about configuring a virtual server, see the ***Configuration Guide for Local Traffic Management***.

To create a virtual server for HTTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_http**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, locate the **HTTP Profile** setting and select **http**.
   This assigns the default HTTP profile to the virtual server.
7. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the HTTP pool you created in the previous section (for example, **http_pool**).
8. From the **Default Persistence Profile** setting, select **source_addr**.
   This implements simple persistence, using the default source address affinity profile.
9. Click **Finished**.

9

Load Balancing HTTP Traffic with Cookie Persistence

- Introducing basic HTTP load balancing
- Configuring HTTP load balancing with cookie persistence

Introducing basic HTTP load balancing

Many computing environments want to use a BIG-IP system to intelligently manage their HTTP traffic. You can easily control your HTTP traffic by implementing a BIG-IP system feature known as an HTTP profile. An HTTP profile is a group of settings that affects the behavior of HTTP traffic. An HTTP profile defines the way that you want the system to manage HTTP traffic.

You can use the default HTTP profile, with all of its default values, or you can create a custom HTTP profile. When you create a custom HTTP profile, you not only modify the setting values, but you can enable more advanced features such as data compression of server responses.

When you configure the BIG-IP system to manage HTTP traffic, you can also implement cookie-based session persistence. ***Cookie persistence*** directs session requests to the same server based on HTTP cookies that the BIG-IP system stores in the client's browser. To implement cookie persistence, the BIG-IP system offers a default persistence profile that you can implement, or you can create a custom cookie persistence profile.

This chapter describes how to set up a basic HTTP load balancing scenario and cookie persistence, using the default HTTP profile and a custom cookie persistence profile. For detailed information on managing HTTP traffic and setting up cookie persistence, see the ***Configuration Guide for Local Traffic Management***.

Configuring HTTP load balancing with cookie persistence

To set up basic HTTP load balancing with persistence that is based on cookies, you need to:

- Create a custom cookie persistence profile
- Create a load balancing pool
- Create a virtual server to process the HTTP traffic and send it to the pool

Because this solution configures HTTP load balancing using the existing default HTTP profile, you do not need to specifically configure a profile for managing HTTP traffic. The only profile you need to configure is the custom cookie persistence profile.

Creating a custom persistence profile

A good way to implement cookie persistence is to create a custom cookie persistence profile. For more information, see the ***Configuration Guide for Local Traffic Management***.

To create a custom cookie persistence profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. On the menu bar, click **Persistence**.
   This displays the list of default persistence profiles.
3. In the upper-right corner of the screen, click **Create**.
   The New Persistence Profile screen opens.
4. In the **Name** box, type a name for the profile, such as **mycookie_profile**.
5. From the **Persistence Type** list, select **Cookie**.
6. From the **Parent Profile** list, select **cookie**.
7. To the far right of the **Cookie Method** setting, check the Custom select box.
8. From the **Cookie Method** list, select **HTTP Cookie Insert**.
9. Leave the **Cookie Name** setting disabled.
10. In the **Expiration** setting, clear the **Session Cookie** check box.
    Additional settings appear.
11. In the **Minutes** box, type **60**.
12. Click **Finished**.

Creating a pool

The next task is to create a load balancing pool to which to load balance HTTP connections. Use the Configuration utility to create this pool. For more detailed information about configuring a pool, see the ***Configuration Guide for Local Traffic Management***.

To create a pool for load balancing HTTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **http_pool**.
4. From the **Health Monitors** list, from the **Available** box select **http**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) In the **Service Port** box, type **80**, or select **HTTP**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
6. Click **Finished**.

Creating a virtual server

The next task in a basic configuration is to define a virtual server that references the HTTP pool. You use the Configuration utility to create the virtual server. For more information about configuring a virtual server, see the ***Configuration Guide for Local Traffic Management***.

To create a virtual server for HTTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_http**.

4. In the **Destination** box:
   a) Verify that the type of virtual server is **Host**
   b) In the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **HTTP Profile** list, select **http**.
   This assigns the default HTTP profile to the virtual server.
8. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the HTTP pool you created in the previous section (for example, **http_pool**).
9. From the **Default Persistence Profile** list, select the name of the custom cookie profile you created earlier, such as **mycookie_profile**.
   This implements cookie persistence, using the custom cookie profile.
10. Click **Finished**.

10

Compressing HTTP Responses

- Introducing HTTP data compression
- Creating a custom HTTP profile
- Creating a virtual server

Introducing HTTP data compression

An optional feature of the BIG-IP system is the system's ability to off-load HTTP compression tasks from the target server. All of the tasks that you need to configure HTTP compression, as well as the compression software itself, are centralized on the BIG-IP system.

The primary way to enable the HTTP compression option is by setting the **Compression** setting of an HTTP profile to **Enabled**. This causes the system to compress HTTP content for any responses matching the values that you specify in the **Request-URI** or **Content-Type** settings of the HTTP profile.

Tip

*If you want to enable HTTP compression for specific connections, you can write an iRule that specifies the **HTTP:compress enable** command. For more information, see the **Configuration Guide for Local Traffic Management**.*

Using the BIG-IP system HTTP compression feature, you can include or exclude certain types of URIs or files that you specify. This is useful because some URI or file types might already be compressed. Using CPU resources to compress already-compressed data is not recommended because the cost of compressing the data usually outweighs the benefits. Examples of regular expressions that you might want to specify for exclusion are **.*\.pdf**, **.*\.gif**, or **.*\.html**.

To configure HTTP data compression, you need to:

- Create a custom HTTP profile
- Create a virtual server to process compressed HTTP responses.

For more detailed, background information on configuring compression and virtual servers, see the ***Configuration Guide for Local Traffic Management***.

Creating a custom HTTP profile

The first task in configuring HTTP data compression on the BIG-IP system is to create a custom HTTP profile. An ***HTTP profile*** defines the way that you want the BIG-IP system to manage HTTP traffic.

After you create the custom HTTP profile, you create a virtual server and assign the custom profile to that virtual server.

To create a custom HTTP profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   This displays a list of any existing HTTP profiles, including the default profile **http**.
2. In the upper-right corner of the screen, click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** box, type a name for the custom profile, such as **http_compress**.
4. Ensure that the **Parent Profile** setting is set to **http**.
5. In the Settings area of the screen, retain all default values.
6. In the Compression area, for the **Compression** setting, on the far right side of the screen, click the Custom box and select **Enabled** from the list.
7. If you want to base compression on URIs specified in the HTTP request headers:
   a) Locate the **URI Compression** setting, click the Select box on the far right of the screen, and select **URI List** from the list.
   This displays the **URI List** settings.
   b) Specify any regular expressions that you want to include or exclude from compression.
   Examples of regular expressions are **.*\.pdf**, **.*\.gif**, or **.*\.html**.
8. If you want to base compression on the type of response content:
   a) Locate the **Content Compression** setting, click the Select box on the far right of the screen, and select **Content List** from the list.
   This displays the **Content List** settings.
   b) Specify values for content you want to include or exclude from compression.
   Examples of content types that you can specify are **application/pdf** and **image/**.**
9. For all other settings in the Compression area of the screen, retain the default values, or configure them to suite your needs.
10. Click **Finished**.

Creating a virtual server

The next task in configuring HTTP compression is to define a virtual server that references the custom HTTP profile that you created in the previous section. You use the Configuration utility to create the virtual server. For more information about configuring a virtual server, see the ***Configuration Guide for Local Traffic Management***.

To create a virtual server for HTTP compression

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_http_compress**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **HTTP Profile** list, select the custom HTTP profile that you created in the previous section. In our example, this value would be **http_compress**.
   This assigns the custom HTTP profile to the virtual server.
8. In the Resources area of the screen, locate the **Default Pool** setting and select a pool name.
9. From the **Default Persistence Profile** list, select **source_addr**.
   This implements the default profile for source address affinity persistence.
10. Click **Finished**.

After you have created a custom HTTP profile and a virtual server, you can test the configuration by attempting to pass HTTP traffic through the virtual server. Check to see that the BIG-IP system includes and excludes the responses that you specified in the custom profile, and that the system compresses the data as specified.

11

Configuring HTTPS Load Balancing

- Introducing HTTPS load balancing
- Creating an SSL key and certificate
- Creating a custom Client SSL profile
- Creating a pool
- Creating a virtual server

Introducing HTTPS load balancing

When you want to load balance HTTPS traffic, you can configure the BIG-IP system to perform the SSL handshaking that target web servers normally perform. A common way to configure the BIG-IP system is to enable it to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client.

In general, the way to configure the BIG-IP system to perform SSSL handshaking (and thus process HTTPS traffic), is to first request and install an SSL key and certificate onto the BIG-IP system. Using the key/certificate pair, the BIG-IP system can act as a server to decrypt the client request before sending it on to the server, and it can encrypt the server response before sending it back to the client.

After installing the key/certificate pair, you can create a custom Client SSL profile. A ***Client SSL profile*** is a type of traffic profile that determines the way that the BIG-IP system processes client requests that are sent by way of a fully SSL-encapsulated protocol (in this case, HTTPS requests).

Next, you must create a pool of servers for load balancing the HTTPS requests.

Finally, you must create a virtual server to process the HTTPS traffic, according to the settings you configured in the custom Client SSL profile.

For more detailed, background information on SSL certificates, SSL profiles, load balancing pools, and virtual servers, see the ***Configuration Guide for Local Traffic Management***.

Creating an SSL key and certificate

Before you can load balance HTTPS traffic, you must create an SSL key and certificate to install onto the BIG-IP system. With an SSL key and certificate, and the custom Client SSL profile that you create next, the BIG-IP system can perform the SSL handshaking normally performed by a target web server.

For purposes of testing that you can pass HTTPS traffic successfully, you can use a self-signed certificate, rather than one signed by a trusted certificate authority.

To create a self-signed key/certificate pair

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SSL Certificates**.
   This displays a list of existing SSL certificates.
2. On the upper-right corner of the screen, click **Create**.
   This opens the New SSL Certificate screen.
3. In the **Name** box, type a name for the certificate, such as **my_cert**.
4. From the **Issuer** list, select **Self.**
5. In the **Common Name** box, type either the IP address for the virtual server you will create later on, or a DNS name that resolves to the virtual server's IP address.
6. In the **Division** box, type your company name.
7. In the **Organization** box, type your department name.
8. In the **Locality** box, type your city name.
9. In the **State or Province** box, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** box, type your email address.
12. In the **Challenge Password** box, type a password.
13. In the **Confirm Password** box, re-type the password you typed in the **Challenge Password** box.
14. In the Key Properties area of the screen, from the **Size** list, select **1024**.
15. Click **Finished**.

Creating a custom Client SSL profile

The second task in configuring HTTPS load balancing on the BIG-IP system is to create a custom Client SSL profile. A ***Client SSL profile*** is a group of settings that enable the BIG-IP system to perform decryption and encryption for client-side SSL traffic. Some of data you specify in the Client SSL profile are the names of the key and certificate you created in the previous section.

After you create the custom Client SSL profile, you create a load balancing pool, and then create a virtual server, assigning the custom profile to that virtual server.

To create a custom Client SSL profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
2. From the SSL menu, choose Client SSL.
   This displays a list of any existing Client SSL profiles, including the default profile **clientssl**.
3. In the upper-right corner of the screen, click **Create**.
   The New Client SSL Profile screen opens.
4. In the **Name** box, type a name for the custom profile, such as **clientssl_profile**.
5. Ensure that the **Parent Profile** setting is set to **clientssl**.
6. For the **Certificate** setting, click the Custom box on the far right side of the screen.
7. From the **Certificate** list, select the name of the certificate you created in the previous section.
   Using our example, this name would be **my_cert.crt**.
8. For the **Key** setting, click the Custom box on the far right side of the screen.
9. From the **Key** list, select the name of the key you created in the previous section.
   Using our example, this name would be **my_cert.key**.
10. Click **Finished**.

Creating a pool

The next task in this process is to create a load balancing pool to load balance HTTPS connections. Use the Configuration utility to create this pool. After you create the pool, you assign it to a virtual server that you create.

To create a pool for load balancing HTTPS traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **https_pool**.
4. For the **Health Monitors** setting, from the **Available** box select **http** or **https_443**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) In the **Service Port** box, type **443**, or select **HTTPS**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
6. Click **Finished**.

Creating a virtual server

The final task in configuring HTTPS load balancing is to define a virtual server that references the custom Client SSL profile and the load balancing pool that you created in previous sections. You use the Configuration utility to create the virtual server.

To create a virtual server for HTTP compression

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_clientssl**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **443**, or select **HTTPS** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **Client SSL Profile** list, select the custom Client SSL profile that you created in a previous section. In our example, this value would be **clientssl_profile**
   This assigns the custom Client SSL profile to the virtual server.
8. In the Resources area of the screen, locate the **Default Pool** setting and select the pool name that you created in a previous section. Using our example, this would be **https_pool**.
9. From the **Default Persistence Profile** list, select **source_addr**.
   This implements the default profile for source address affinity persistence.
10. Click **Finished**.

After you have created an SSL key/certificate pair, a custom Client SSL profile, a load balancing pool, and a virtual server, you can test the configuration by attempting to pass HTTPS traffic through the virtual server.

12

Configuring HTTPS Load Balancing with Data Compression

- Introducing HTTPS load balancing with compression
- Creating an SSL key and certificate
- Creating a custom Client SSL profile
- Creating a custom HTTP profile for compression
- Creating a pool
- Creating a virtual server

Introducing HTTPS load balancing with compression

When you want to load balance HTTPS traffic, you can configure the BIG-IP system to perform the SSL handshaking that target web servers normally perform. A common way to configure the BIG-IP system is to enable it to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client.

In general, the way to configure the BIG-IP system to perform SSL handshaking (and thus process HTTPS traffic), is to first request and install an SSL key and certificate onto the BIG-IP system. Using the key/certificate pair, the BIG-IP system can act as a server to decrypt the client request before sending it on to the server, and it can encrypt the server response before sending it back to the client.

After installing the key/certificate pair, you can create a custom Client SSL profile. A ***Client SSL profile*** is a type of traffic profile that determines the way that the BIG-IP system processes client requests that are sent by way of a fully SSL-encapsulated protocol (in this case, HTTPS requests).

Next, you must create a pool of servers for load balancing the HTTPS requests.

Finally, you must create a virtual server to process the HTTPS traffic, according to the settings you configured in the custom Client SSL profile.

For more detailed, background information on SSL certificates, SSL profiles, load balancing pools, and virtual servers, see the ***Configuration Guide for Local Traffic Management***.

Creating an SSL key and certificate

Before you can load balance HTTPS traffic and enable compression, you must create an SSL key and certificate to install onto the BIG-IP system. With an SSL key and certificate, and the custom Client SSL profile that you create next , the BIG-IP system can perform the SSL handshaking normally performed by a target web server.

For purposes of testing that you can pass HTTPS traffic successfully, you can use a self-signed certificate, rather than one signed by a trusted certificate authority.

To create a self-signed key/certificate pair

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SSL Certificates**.
   This displays a list of existing SSL certificates.
2. On the upper-right corner of the screen, click **Create**.
   This opens the New SSL Certificate screen.
3. In the **Name** box, type a name for the certificate, such as **my_cert**.
4. From the **Issuer** list, select **Self.**
5. In the **Common Name** box, type either the IP address for the virtual server you will create later on in this chapter, or a DNS name that resolves to the virtual server's IP address.
6. In the **Division** box, type your company name.
7. In the **Organization** box, type your department name.
8. In the **Locality** box, type your city name.
9. In the **State or Province** box, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** box, type your email address.
12. In the **Challenge Password** box, type a password.
13. In the **Confirm Password** box, re-type the password you typed in the **Challenge Password** box.
14. In the Key Properties area of the screen, from the **Size** list, select **1024**.
15. Click **Finished**.

Creating a custom Client SSL profile

The next task in configuring HTTPS load balancing with compression is to create a custom Client SSL profile. A ***Client SSL profile*** is a group of settings that enable the BIG-IP system to perform decryption and encryption for client-side SSL traffic. Some of data you specify in the Client SSL profile are the names of the key and certificate you created in the previous section.

After you create the custom Client SSL profile, you create a load balancing pool, and then create a virtual server, assigning the custom profile to that virtual server.

To create a custom Client SSL profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the SSL menu, choose Client SSL.
   This displays a list of any existing Client SSL profiles, including the default profile **clientssl**.
3. In the upper-right corner of the screen, click **Create**.
   The New Client SSL Profile screen opens.
4. In the **Name** box, type a name for the custom profile, such as **clientssl_profile**.
5. Ensure that the **Parent Profile** setting is set to **clientssl**.
6. For the **Certificate** setting, click the Custom box on the far right side of the screen.
7. From the **Certificate** list, select the name of the certificate you created in the previous section.
   Using our example, this name would be **my_cert.crt**.
8. For the **Key** setting, click the Custom box on the far right side of the screen.
9. From the **Key** list, select the name of the key you created in the previous section.
   Using our example, this name would be **my_cert.key**.
10. Click **Finished**.

Creating a custom HTTP profile for compression

To enable HTTP data compression on the BIG-IP system, you must create a custom HTTP profile. An ***HTTP profile*** defines the way that you want the BIG-IP system to manage HTTP traffic.

After you create the custom HTTP profile, you create a load balancing pool. Then you create a virtual server, assigning the custom HTTP profile to that virtual server.

To create a custom HTTP profile for compression

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   This displays a list of any existing HTTP profiles, including the default profile **http**.
2. In the upper-right corner of the screen, click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** box, type a name for the custom profile, such as **http_compress**.
4. Ensure that the **Parent Profile** setting is set to **http**.
5. In the Settings area of the screen, retain all default values.
6. For the **Compression** setting, on the far right side of the screen, click the Select box and select **Enabled** from the list.
7. If you want to base compression on URIs specified in the HTTP request headers:
   a) Locate the **URI Compression** setting, click the Select box on the far right of the screen, and select **URI List** from the list.
   This displays the U**RI List** settings.
   b) Specify any regular expressions that you want to include or exclude from compression.
   Examples of regular expressions are **.*\.pdf**, **.*\.gif**, or **.*\.html**.
8. If you want to base compression on the type of response content:
   a) Locate the **Content Compression** setting, click the Select box on the far right of the screen, and select **Content List** from the list.
   This displays the **Content List** settings.
   b) Specify values for content you want to include or exclude from compression.
   Examples of content types that you can specify are **application/pdf** and **image/**.**
9. For all other settings in the Compression area of the screen, retain the default values, or configure them to suite your needs.
10. Click **Finished**.

Creating a pool

The next task in the process is to create a load balancing pool to load balance HTTPS connections. Use the Configuration utility to create this pool. After you create the pool, you assign it to a virtual server that you create.

To create a pool for load balancing HTTPS traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **https_pool**.
4. For the **Health Monitors** setting, from the **Available** box select **https** or **https_443**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. In the Resource area, for the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) In the **Service Port** box, type **443**, or select **HTTPS**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
6. Click **Finished**.

Creating a virtual server

The final task in configuring HTTPS load balancing is to define a virtual server that references the custom Client SSL profile and the load balancing pool that you created in previous sections. You use the Configuration utility to create the virtual server.

To create a virtual server for HTTP compression

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_clientssl**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **443**, or select **HTTPS** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **HTTP Profile** list, select the name of the HTTP profile that you created.
   Using our example, this value would be **http_compress**.
8. From the **Client SSL Profile** list, select the custom Client SSL profile that you created in a previous section. In our example, this value would be **clientssl_profile**
   This assigns the custom Client SSL profile to the virtual server.
9. In the Resources area of the screen, locate the **Default Pool** setting and select the pool name that you created in a previous section. Using our example, this would be **https_pool**.
10. From the **Default Persistence Profile** list, select **source_addr**. This implements the default profile for source address affinity persistence.
11. Click **Finished**.

You can now test the configuration by attempting to pass HTTPS traffic through the virtual server. Check to see that the BIG-IP system includes and excludes the responses that you specified in the custom HTTP profile, and that the system compresses the data as specified.

13

Using RAM Cache for HTTP Traffic

- Introducing HTTP RAM Cache
- Creating a custom HTTP profile
- Creating a virtual server

Introducing HTTP RAM Cache

An optional feature of the BIG-IP system is its HTTP RAM cache. A ***RAM cache*** is a cache of HTTP objects stored in the BIG-IP system's RAM that subsequent connections can reuse to reduce the amount of load on the back-end servers.

When to use the RAM Cache

The RAM Cache feature provides the ability to reduce the traffic load to back-end servers. This ability is useful if an object on a site is under high demand, if the site has a large quantity of static content, or if the objects on the site are compressed.

- **High demand objects**
  This feature is useful if a site has periods of high demand for specific content. With RAM Cache configured, the content server only has to serve the content to the BIG-IP system once per expiration period.
- **Static content**
  This feature is also useful if a site consists of a large quantity of static content such as CSS, javascript, or images and logos.
- **Content compression**
  For compressible data, the RAM Cache can store data for clients that can accept compressed data. When used in conjunction with the compression feature on the BIG-IP system, the RAM Cache takes stress off of the BIG-IP system and the content servers.

The items you can cache

The RAM Cache feature is fully compliant with the cache specifications described in RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*. This means that you can configure RAM Cache to cache the following content types:

- 200, 203, 206, 300, 301, and 410 responses.
- Responses to GET methods by default.
- Other HTTP methods for URIs specified in the URI Include list or specified in an iRule.
- Content based on the User-Agent and Accept-Encoding values. The RAM Cache holds different content for **Vary** headers.

To use the RAM Cache feature, you need to:

- Create a custom HTTP profile.
- Create a virtual server.

For more detailed, background information on configuring the RAM Cache feature, see the ***Configuration Guide for Local Traffic Management***.

Creating a custom HTTP profile

The first task in configuring the HTTP RAM Cache feature on the BIG-IP system is to create a custom HTTP profile. An ***HTTP profile*** defines the way that you want the BIG-IP system to manage HTTP traffic.

After you create the custom HTTP profile, you create a virtual server and assign the custom profile to that virtual server.

To create a custom HTTP profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   This displays a list of any existing HTTP profiles, including the default profile **http**.
2. In the upper-right corner of the screen, click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** box, type a name for the custom profile, such as **http_ramcache**
4. Ensure that the **Parent Profile** setting is set to **http**.
5. In the Settings area of the screen, retain all default values.
6. Scroll down to the RAM Cache area of the screen.
7. For the **RAM Cache** setting, on the far right side of the screen, click the Select box, and select **Enabled** from the **RAM Cache** list.
8. For all other settings in the RAM Cache area of the screen, retain the default values, or configure them to suit your needs.
9. Click **Finished**.

Creating a virtual server

The next task in configuring the RAM Cache feature is to define a virtual server that references the custom HTTP profile that you created in the previous section.You use the Configuration utility to create the virtual server.

To create a virtual server for HTTP RAM Cache

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_http_compress**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **HTTP Profile** list, select the custom HTTP profile that you created in the previous section. In our example, this value would be **http_ramcache**.
   This assigns the custom HTTP profile to the virtual server.
8. In the Resources area of the screen, locate the **Default Pool** setting and select a pool name.
9. From the **Default Persistence Profile** list, select **source_addr**.
   This implements the default profile for source address affinity persistence.
10. Click **Finished**.

14

Load Balancing Passive Mode FTP Traffic

- Introducing FTP load balancing
- Creating a custom FTP monitor
- Creating a pool
- Creating a virtual server

Introducing FTP load balancing

You can set up the BIG-IP system to load balance passive mode FTP traffic. To do this, you create the following:

- A custom FTP health monitor
- A pool for load balancing FTP traffic
- A virtual server for processing FTP traffic

When you create the virtual server, you can configure it to use the default FTP profile. An ***FTP profile*** determines the way that the BIG-IP system processes FTP traffic.

This chapter describes how to create the objects listed above, using the default FTP profile. For more detailed information on managing FTP traffic, see the ***Configuration Guide for Local Traffic Management***.

Creating a custom FTP monitor

Using the Configuration utility, you can create a custom FTP monitor to monitor files on your FTP server.

To create a custom FTP monitor

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   This displays a list of existing health and performance monitors.
2. On the upper-right corner of the screen, click **Create**.
   This opens the New Monitor screen.
3. In the **Name** box, type a name for the custom monitor, such as **my_ftp_monitor**.
4. From the **Type** list, select **FTP**.
   This displays additional FTP monitor settings.
5. In the **User Name** box, type the login name for the FTP server.
6. In the **Password** box, type the password for the login name.
7. In the **Path/Filename** box, type the path and name for the file you want to monitor.
8. Verify that the **Mode** setting is set to **Passive**.
9. For all other settings, retain the default values.
10. Click **Finished**.

After you have created a custom FTP monitor, you can create a load balancing pool for your FTP traffic.

Creating a pool

To load balance passive mode FTP traffic, you create a load balancing pool. When you create the pool, you assign the custom FTP monitor that you created in the previous section.

After creating the pool, you assign it to the virtual server that you create.

To create a pool for load balancing FTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **ftp_pool**.
4. For the **Health Monitors** setting, from the **Available** box select the name of the custom FTP monitor, such as **my_ftp_monitor**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. In the Resources section, ensure that the **Load Balancing Method** setting is set to **Round Robin.**
6. Ensure that **Priority Group Activation** is set to **Disabled**.
7. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) From the **Service Port** list, select **FTP**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
8. Click **Finished**.

Creating a virtual server

The next task in a basic configuration is to define a virtual server that references the FTP profile and the FTP pool. You use the Configuration utility to create the virtual server.

To create a virtual server for FTP traffic

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_ftp**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. From the **Service Port** list, select **FTP**.
6. From the **FTP Profile** list, select **ftp**.
   This assigns the default FTP profile to the virtual server.
7. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the FTP pool you created in the previous section (for example, **ftp_pool**).
8. From the **Default Persistence Profile** setting, select **source_addr**.
   This implements simple persistence, using the default source address affinity profile.
9. Click **Finished**.

15

Load Balancing Passive Mode FTP Traffic with Rate Shaping

- Introducing FTP load balancing with rate shaping with rate shaping
- Creating a custom FTP monitor
- Creating a pool
- Creating a rate class
- Creating a virtual server

Introducing FTP load balancing with rate shaping

You can set up the BIG-IP system to load balance passive mode FTP traffic with rate shaping. To do this, you create the following:

- A custom FTP health monitor
- A pool for load balancing FTP traffic
- Create a rate class
- A virtual server for processing FTP traffic

When you create the virtual server, you can configure it to use the default FTP profile. An ***FTP profile*** determines the way that the BIG-IP system processes FTP traffic.

This chapter describes how to create the objects listed above, using the default FTP profile. For more detailed information on managing FTP traffic, see the ***Configuration Guide for Local Traffic Management***.

Note that the rate shaping feature is optional on the BIG-IP system. Therefore, you must have purchased a license for the rate shaping feature before you can use rate shaping to control the load balancing of passive FTP traffic.

Creating a custom FTP monitor

Using the Configuration utility, you can create a custom FTP monitor to monitor files on your FTP server.

To create a custom FTP monitor

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   This displays a list of existing health and performance monitors.
2. On the upper-right corner of the screen, click **Create**.
   This opens the New Monitor screen.
3. In the **Name** box, type a name for the custom monitor, such as **my_ftp_monitor**.
4. From the **Type** list, select **FTP**.
   This displays additional FTP monitor settings.
5. In the **User Name** box, type the login name for the FTP server.
6. In the **Password** box, type the password for the login name.
7. In the **Path/Filename** box, type the path and name for the file you want to monitor.
8. Verify that the **Mode** setting is set to **Passive**.
9. For all other settings, retain the default values.
10. Click **Finished**.

After you have created a custom FTP monitor, you can create a load balancing pool for your FTP traffic.

Creating a pool

To load balance passive mode FTP traffic, you create a load balancing pool. When you create the pool, you assign the custom FTP monitor that you created in the previous section.

After creating the pool, you assign it to the virtual server that you create in the next section.

To create a pool for load balancing FTP traffic with rate shaping

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **ftp_pool**.
4. For the **Health Monitors** setting, from the **Available** box select the name of the custom FTP monitor, such as **my_ftp_monitor**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. In the Resources section, ensure that the **Load Balancing Method** setting is set to **Round Robin.**
6. Ensure that the **Priority Group Activation** setting is set to **Disabled**.
7. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) From the **Service Port** list, select **FTP**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
8. Click **Finished**.

Creating a rate class

To implement rate shaping, you create a rate class. By creating a rate class, you can load balance passive mode FTP traffic that is controlled by rate shaping.

◆ Note

Rate shaping is an optional feature of the BIG-IP system. Before attempting to implement rate shaping, verify that you are licensed to use the feature.

To create a rate class

1. On the Main tab of the navigation pane, expand **Local Traffic** and click **Rate Shaping**.
   The Rate Shaping screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Rate Class screen opens.
3. In the **Name** box, type a name for the rate class, such as **ftp_rateclass**.
4. In the **Base Rate** box, type **1** and select **Mbps** from the list.
5. In the **Ceiling Rate** box, type **10** and select **Mbps** from the list.
6. In the **Burst Rate** box, type **10000**.
7. For all other settings, retain the default values.
8. Click **Finished**.

Creating a virtual server

The next task in a basic configuration is to define a virtual server that references the FTP profile and the FTP pool. You use the Configuration utility to create the virtual server.

To create a virtual server for FTP traffic with rate shaping

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_ftp**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. From the **Service Port** list, select **FTP**.
6. From the **FTP Profile** list, select **ftp**.
   This assigns the default FTP profile to the virtual server.
7. From the **Rate Class** list, select the name of the rate class you created in the previous section (for example, **ftp_rateclass**).
8. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the FTP pool you created in the previous section (for example, **ftp_pool**).
9. From the **Default Persistence Profile** setting, select **source_addr**.
   This implements simple persistence, using the default source address affinity profile.
10. Click **Finished**.

16

Setting up a One-IP Network Topology

- Introducing the one-IP network topology
- Creating a pool for a one-IP network topology
- Creating a virtual server
- Defining a default route
- Configuring a client SNAT

Introducing the one-IP network topology

Another configuration option you can use with the BIG-IP system is a one-IP network topology. This differs from the typical two-network configuration in two ways:

- Because there is only one physical network, this configuration does not require more than one interface on the BIG-IP system.
- Clients need to be assigned SNATs to allow them to make connections to servers on the network in a load balancing pool.

The single interface configuration is shown in Figure 16.1.

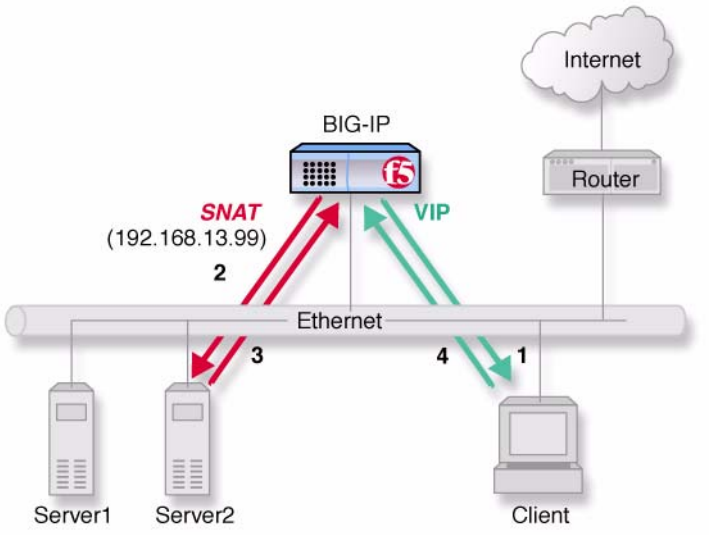

Figure 16.1 *An example of a single interface topology*

To set up this configuration, you need to complete the following tasks on the BIG-IP system:

- Create a load balancing pool for the content servers.
- Create a virtual server to load balance traffic to the content server pool.
- Define a default route for the external VLAN.
- Configure a SNAT for the client.

Creating a pool for a one-IP network topology

The first task required to set up this solution is to create a pool that contains the content servers that you want to load balance. Before creating the pool, verify that all content servers for the pool are in the network of VLAN **external.**

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. From the Configuration list, select **Advanced**.
4. In the **Name** box, type a name for the pool, such as **server_pool**.
5. For the **Health Monitors** setting, from the **Available** box select **http**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
6. For the **Allow SNAT** setting, verify that the value is **Yes**.
7. For the remaining settings in the Configuration area of the screen, retain the default values.
8. In the Resources area of the screen, use the default values for the **Load Balancing Method** and **Priority Group Activation** settings.
9. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) In the **Service Port** box, type **80**, or select **HTTP**.
   d) Click **Add.**
   e) Repeat steps b, c, and d for each server in the pool.
10. Click **Finished**.

Creating a virtual server

The second task required to set up this solution is to create a virtual server that references the pool of servers that you want to load balance. The pool that the virtual server references is the pool you created in the previous step.

To create a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_one_ip**.
4. For the **Destination** setting:
   a) Verify that the type of virtual server is **Host**
   b) In the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, retain the value of the **Protocol** setting, **TCP**.
7. From the **HTTP Profile** list, select **http**.
   This assigns the default HTTP profile to the virtual server.
8. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the pool you created in the previous section (using our example, this would be **server_pool**).
9. Click **Finished**.

Defining a default route

Another task that you must perform to implement one-IP network load balancing is to define a default route for the VLAN **external.**

To define a default route

1. On the Main tab of the navigation pane, expand **Network** and click **Routes**
   The Routes screen opens.
2. In the upper-right corner of the screen, click **Add**.
   The New Route screen opens.
3. For the **Type** setting, verify that it is set to **Default Gateway**.
   This disables the **Destination** and **Netmask** settings.
4. For the **Resource** setting:
   a) From the list on the left, select **Use VLAN**.
   b) From the list on the right, select **external**.
5. Click **Finished.**

Configuring a client SNAT

Finally, configure the BIG-IP system to handle connections originating from the client. You must define a SNAT in order to change the source address on the packet to the SNAT external address, which is located on the BIG-IP system. Otherwise, if the source address of the returning packet is the IP address of the content server, the client does not recognize the packet because the client sent its packets to the IP address of the virtual server, not the content server.

If you do not define a SNAT, the server returns the packets directly to the client without giving the BIG-IP system the opportunity to translate the source address from the server address back to the virtual server. If this happens, the client might reject the packet as unrecognizable.

To configure a client SNAT

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SNATs.**
   The SNATs screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New SNAT screen opens.
3. In the **Name** box, type a name for the SNAT, such as **snat_one_ip**.
4. In the **Translation** box, type an IP address that you want to use as a translation IP address.
5. From the **Origin** list, select **Address List**.
   This displays additional configuration settings.
6. For the **Address List** setting:
   a) For the **Type** setting, verify that **Host** is enabled.
   b) In the **Address** box, type a client IP address.
   c) Click **Add**.
   d) Repeat this process for each client to which you want to assign the translation address.
7. From the **VLAN Traffic** list, select **Enabled on**.
8. For the **VLAN List** setting, from the **Available** box select **external**, and click the Move button (**<<**) to move the VLAN name to the **Active** box.
9. Click **Finished**.

17

Using Link Aggregation with Tagged VLANs

- Introducing link aggregation with tagged VLAN interfaces
- Using the two-network aggregated tagged interface topology
- Using the one-network aggregated tagged interface topology

Introducing link aggregation with tagged VLAN interfaces

You can use the BIG-IP system in an aggregated two-interface load balancing topology. This topology contains two tagged interfaces (links), 4.1 and 5.1, aggregated together. There are two VLANs, **VLAN1** and **VLAN2**, passing traffic to and from the switch. A virtual server on VLAN2 load balances connections to the servers on VLAN2.

Thus, both links are on both VLANs, and inbound and outbound traffic can use either interface.

Aggregating the two links has two advantages:

- It increases the bandwidth of the individual network interface cards (NICs) in an additive manner.
- If one link goes down, the other link can handle the traffic by itself.

You can use link aggregation in two configurations, the two-network configuration and the single-network configuration.

Using the two-network aggregated tagged interface topology

Figure 17.1 shows a two-IP network topology, with one network connected to the external VLAN, and a separate network connected to the internal VLAN.

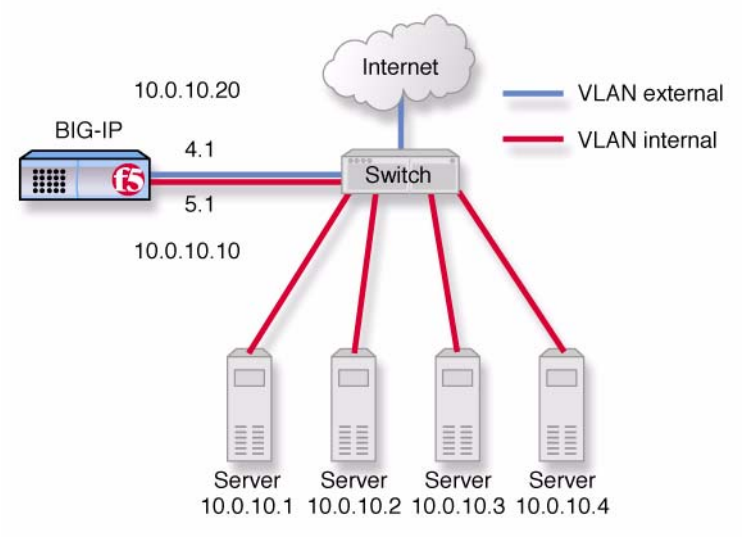

Figure 17.1 *An example of an aggregated two-interface load balancing configuration*

To configure the BIG-IP system for the two-network solution, you must complete the following tasks:

- Create a trunk to aggregate the links.
- Add tagged interfaces to the internal and external VLANs.
- Create a pool of web servers that you want to load balance.
- Create a virtual server that load balances the web servers.

◆ Note

*This example assumes that you are using the default **internal** and **external** VLAN configuration. It also assumes that the self IP addresses on each VLAN are on the same IP network as the BIG-IP system.*

Aggregating the links

The first task for this solution is to aggregate the links. To do this, you must create a trunk, assign interfaces to the trunk as members, and then enable Link Aggregation Control Protocol (LACP).

To aggregate links

1. On the Main tab of the navigation pane, expand **Network**, and click **Trunks**.
   The Trunks screen opens.
2. On the upper-right corner of the screen, click **Create**.
   The New Trunk screen opens.
3. In the **Name** box, type a name for the trunk, such as **trunk1**.
4. For the **Interfaces** setting, locate the **Available** box and select an interface.

   Note: *The lowest-numbered interface is the controlling, or reference, link.*
5. Using the Move button, move the interface number to the **Members** box.
6. Repeat step 5 for all interfaces that you want to include as trunk members.
7. For the **LACP** setting, check the box.
   This enables link aggregation.
8. Click **Finished**.

Adding tagged interfaces to VLANs

After you aggregate the links, you can create the VLAN tags. Creating VLAN tags means specifying the interfaces assigned to a VLAN as **tagged** interfaces.

WARNING

You should perform this task from the console. If you attempt to change the tags from a remote workstation, you will be disconnected.

To add tagged interfaces to an existing VLAN

1. On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLAN screen opens.
2. In the Name column, click the VLAN name **internal.**
   This displays the properties of that VLAN.
3. For the **Interfaces** setting, locate the **Available** box and select the name of an interface on the network.

4. Click the Move button to move the interface name to the **Tagged** box.
   This assigns the selected interface to the VLAN, as a tagged interface.
5. Click **Update**.
6. Return to the list of existing VLANs.
7. Using the same process that you used for the VLAN **internal**, add tagged interfaces for the VLAN **external**.
8. Click **Update**.

Creating a pool of web servers to load balance

After you create the network environment for the BIG-IP system, you can create the pool of web servers you want to load balance.

To create a pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **myweb_pool**.
4. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a web server in the pool.
   c) From the **Service Port** list, select a service.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
5. Click **Finished**.

Creating a virtual server to load balance the web servers

After you create the pool of web servers you want to load balance, you can create the virtual server.

To create a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.

2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_myweb**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server. Using the example in Figure 17.1, on page 17-2, this address could be **10.0.10.30**.
5. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the pool you created in the previous section (for example, **myweb_pool**).
6. From the **Default Persistence Profile** setting, select **source_addr**. This implements simple persistence, using the default source address affinity profile.
7. Click **Finished**.

Using the one-network aggregated tagged interface topology

Figure 17.2 shows a single IP network topology. The one-network topology is identical to the two-network topology in all respects except that in the one-network solution, the internal and external VLANs connect to members of the same IP network. This requires that the two VLANs be grouped in order to be able to exchange packets directly.

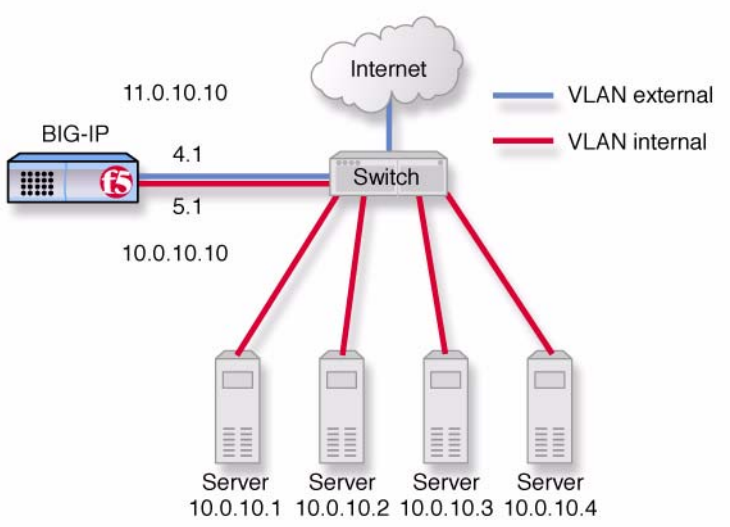

Figure 17.2 *An example of an aggregated two interface load balancing configuration with one IP network*

You configure the one-network topology in exactly the same way as the two-network topology (allowing for the fact that the virtual server address will now belong to the same network as the servers), with one additional step: the internal and external VLANs need to be grouped. Therefore, to configure the BIG-IP system for this solution, you must complete the following tasks:

- Configure the tagged interfaces, load balancing pool, virtual server, and trunk exactly as in the two-network configuration. For more information, see *Using the two-network aggregated tagged interface topology*, on page 17-2.
- Remove the self IP addresses from the internal and external VLANs.
- Combine the internal and external VLANs into a VLAN group.
- Assign a self IP address to the VLAN group.

Removing the self IP addresses from the VLANs

Before you can create a VLAN group, you must remove the self IP addresses from the individual VLANs. After you create the VLAN group, you create a self IP address for the VLAN group, for routing purposes. The individual VLANs no longer need their own self IP addresses.

WARNING

We recommend that you perform this step from the console or from a self IP address you are not going to delete. If you are connected from a remote workstation though a self IP address you are going to delete, you will be disconnected when you delete it.

To remove the self IP addresses from the default VLANs

1. On the Main tab of the navigation pane, expand **Network**, and click **Self IPs**.
   The Self IPs screen opens.
2. Using the IP Address and VLANs columns, locate the self IP addresses for the internal and external VLANs.
3. To the left of each self IP address you want to delete, check the Select box.
4. Click **Delete**.
   A confirmation screen appears.
5. Click **Delete** again.

Creating a VLAN group

Create a VLAN group that includes the internal and external VLANs. Packets received by a VLAN in the VLAN group are copied onto the other VLAN. This allows traffic to pass through the BIG-IP system on the same IP network.

Tip

A VLAN group name can be used anywhere that a VLAN name can be used.

To create a VLAN group

1. On the Main tab of the navigation pane, expand **Network**, and click **VLANs**.
   The VLANs screen opens.
2. From the VLAN Groups menu, choose List.
   This opens the VLAN Groups screen.
3. In the upper-right corner of the screen, click **Create**.
   This opens the New VLAN Group screen.
4. In the **Name** box, type the name **myvlangroup**.

5. For the **VLANs** setting, use the Move button to move the **internal** and **external** VLAN names from the **Available** box to the **Members** box.
6. Click **Finished**.

Creating a self IP for the VLAN group

After you have created the VLAN group, create a self IP address for the VLAN group. The self IP address for the VLAN group provides a route for packets destined for the network. With the BIG-IP system, the path to an IP network is a VLAN. However, with the VLAN group feature used in this example, the path to the IP network **10.0.0.0** is actually through more than one VLAN. Since IP routers are designed to have only one physical route to a network, a routing conflict can occur. The self IP address feature on the BIG-IP system allows you to resolve the routing conflict by putting a self IP address on the VLAN group.

To create a self IP address for a VLAN group

1. On the Main tab of the navigation pane, expand **Network,** and click **Self IPs**.
   The Self IPs screen opens.
2. In the upper-right corner of the screen, click **Create**.
3. In the **IP Address** box, type a self IP address for the VLAN group.
4. In the **Netmask** box, type a netmask for the self IP address.
5. For the **VLAN** setting, select the name **myvlangroup** from the list.
6. Click **Finished**.

18

Setting Up Packet Filtering

- Introducing packet filtering
- Configuring packet filtering

Introducing packet filtering

Packet filters enhance network security by specifying whether a BIG-IP system interface should accept or reject certain packets based on criteria that you specify. Packet filters enforce an access policy on incoming traffic. They apply to incoming traffic only.

You implement packet filtering by creating packet filter rules, using the Configuration utility. The primary purpose of a ***packet filter rule*** is to define the criteria that you want the BIG-IP system to use when filtering packets. Examples of criteria that you can specify in a packet filter rule are:

- The source IP address of a packet
- The destination IP address of a packet
- The destination port of a packet

You specify the criteria for applying packet filter rules within an expression. When creating a packet filter rule, you can instruct the Configuration utility to build an expression for you, in which case you need only choose the criteria from predefined lists, or you can write your own expression text, using the syntax of the **tcpdump** utility. For more information on the **tcpdump** utility, see the online man page for the **tcpdump** command.

◆ Note

Packet filter rules are unrelated to iRules™.

You can also configure global packet filtering that applies to all packet filter rules that you create. The following sections describe how to set global packet filtering options, and how to create and manage individual packet filters rules.

By setting up some basic IP routing and configuring packet filtering, specific hosts on the internal VLAN can connect to the internal VLAN's self IP address. These hosts can also use common Internet services such as HTTP, HTTPS, DNS, FTP, and SSH. Traffic from all other hosts in the internal VLAN is rejected.

To configure this solution, you must:

- Create a SNAT.
- Create a pool of routers (also known as a gateway pool).
- Create a forwarding virtual server.
- Create a packet filter rule.

Configuring packet filtering

This section describes each of the tasks that you need to perform to fully configure packet filtering.

Creating a SNAT

The first task in implementing packet filtering is to create a SNAT.

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **SNATs.**
   The SNATs screen opens.
2. In the upper-right corner, click **Create**.
   The New SNAT screen opens.
3. In the **Name** box, type a unique name for the SNAT.
4. From the **Translation** list, select **Automap.**
5. From the **VLAN Traffic** list, select **Enabled On**.
   This displays the **VLAN List** setting.
6. For the **VLAN List** setting, from the **Available** box select **internal** and **external**, and click the Move button (**<<**) to move the VLAN names to the **Selected** box.
7. Click **Finished.**

Creating a gateway pool

The next task is to define a pool of routers.

To create a gateway pool

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **gateway_pool**.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members.
   The members you add are router IP addresses.
5. Click **Finished**.

Creating a virtual server

The next task is to create a forwarding virtual server that references the pool **gateway_pool**.

To create the virtual servers

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_packetfilter**.
4. For the **Destination** setting:
   a) For **Type**, select **Network**.
   b) In the **Address** box, type the IP address **0.0.0.0**.
   c) In the **Mask** box, type the netmask **0.0.0.0**.
5. From the **Service Port** list, select ***All Ports**
6. In the Configuration area of the screen, locate the **Type** setting and select **Forwarding (IP)**.
7. From the **Protocol** list, select ***All Protocols**.
8. From the **VLAN Traffic** list, select **Enabled On**.
9. For the **VLAN List** setting, from the **Available** box select **internal**, and click the Move button (**<<**) to move the VLAN name to the **Selected** box.
10. In the Resources area of the screen, locate the **Default Pool** setting and select the pool you created previously (**gateway_pool)**.
11. Click **Finished**.

Creating a packet filter rule

The final task in implementing packet filtering is to create a packet filter rule. Note that a packet filter rule is different from an iRule.

To create a packet filter rule

1. On the Main tab of the navigation pane, expand **Network**, and click **Packet Filters**.
   This displays the setting to enable or disable packet filtering, as well as some global packet filter settings.
2. On the menu bar, click **Rules**.
   This displays a list of any existing packet filter rules.

3. From the Packet Filtering list, select **Enabled.**
   This displays additional settings.
4. From the **Unhandled Packet Action** list, select **Accept.**
5. Click **Update**.
6. On the menu bar, click **Rules**.
   This displays a list of existing packet filter rules, if any.
7. On the upper right corner of the screen, click **Create**.
   The New Packet Filter Rule screen opens.
8. In the **Name** box, type a packet filter name, such as **pf_internal**.
9. From the **Order** list, select **First**.
10. From the **Action** list, select **Reject**.
11. From the **Apply to VLAN** list, select **internal**.
12. From the **Logging** list, select **Enabled**.
13. From the **Filter Expression Method** list, select **Enter Expression Text**.
    This displays the **Filter Expression** text box.
14. In the text box, type an expression. For example:

    ```
    not dst port 80 and not dst port 443 and not dst port 53
    and no dst port 22 and not dst port 20 and not dst port
    21 and not dst host <internal self IP address>
    ```

    Note: *Replace* ***<internal self IP address>*** *with the actual self IP address of VLAN* ***internal****. Also, see the* ***tcpdump*** *man page for general information about building expresssions.*
15. Click **Finished**.

19

Implementing Health and Performance Monitors

- Introducing health and performance monitors
- Creating a custom monitor
- Creating a pool
- Creating a virtual server

Introducing health and performance monitors

You can set up the BIG-IP system to monitor the health or performance of certain nodes or servers that are members of a load balancing pool. ***Monitors*** verify connections on pool members and nodes. A monitor can be either a health monitor or a performance monitor, designed to check the status of a pool, pool member, or node on an ongoing basis, at a set interval. If a pool member or node being checked does not respond within a specified timeout period, or the status of a pool member or node indicates that performance is degraded, the LTM system can redirect the traffic to another pool member or node.

Some monitors are included as part of the BIG-IP system, while other monitors are user-created. Monitors that the BIG-IP system provides are called ***pre-configured monitors***. User-created monitors are called ***custom monitors***. For more information on pre-configured and custom monitors, see the ***Configuration Guide for Local Traffic Management***.

Before configuring and using monitors, it is helpful to understand some basic concepts regarding monitor types, monitor settings, and monitor implementation.

- **Monitor types**
  Every monitor, whether pre-configured or custom, is a certain type of monitor. Each type of monitor checks the status of a particular protocol, service, or application. For example, one type of monitor is HTTP. An HTTP type of monitor allows you to monitor the availability of the HTTP service on a pool, pool member, or node. A WMI type of monitor allows you to monitor the performance of a pool, pool member, or node that is running the Windows Management Instrumentation (WMI) software. An ICMP type of monitor simply determines whether the status of a node is **up** or **down**.
- **Monitor settings**
  Every monitor consists of settings with values. The settings and their values differ depending on the type of monitor. In some cases, the BIG-IP system assigns default values. For example, Figure 19.1 shows the settings and default values of an ICMP-type monitor.

```
Name my_icmp
Type ICMP
Interval 5
Timeout 16
Transparent No
Alias Address * All Addresses
```

Figure 19.1 *Example of a monitor with default values*

To implement a health monitor, you complete these tasks:

- Create a custom monitor or decide to use a pre-configured monitor.
- Create a pool for load balancing traffic, and assign a monitor to the pool.
- Create a virtual server for processing traffic.

The remainder of this chapter describes how to create these objects. For more detailed information on implementing monitors, see the ***Configuration Guide for Local Traffic Management***.

◆ Note

If you want to monitor the performance of a RealNetworks® RealServer server or a Windows® server equipped with Windows Management Instrumentation (WMI), you must first download a special plug-in file onto the BIG-IP system. For more information, see Appendix A of the ***Configuration Guide for Local Traffic Management***.

Creating a custom monitor

When you want to monitor a node, a server, or a pool of servers, you can use a pre-configured monitor, or you create a custom monitor. The following procedure describes how to create a custom monitor. If you want to use a pre-configured monitor, you can skip this procedure and move on to the next section, *Creating a pool*, on page 19-4.

To create a custom monitor

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Monitors**.
   This displays a list of existing health and performance monitors.
2. On the upper-right corner of the screen, click **Create**.
   This opens the New Monitor screen.
3. In the **Name** box, type a name for the custom monitor.
   For example, if you are creating a custom HTTP monitor, you can assign the name **my_http_monitor**.
4. From the **Type** list, select the type of monitor you want to create.
   This displays additional monitor settings for you to configure.
5. Change the values of any monitor settings to suit your needs.
6. Click **Finished**.

After you have created a custom monitor, you can create a load balancing pool and assign the monitor name to the pool.

Creating a pool

When you create the pool to load balance traffic, you assign the custom monitor that you created in the previous section to a load balancing pool. Then, after creating the pool, you assign it to the virtual server that you create in the next section.

Assigning a monitor to a pool

One way to assign a monitor is to create the pool and assign the monitor to the pool itself. When you assign a monitor to a pool, all members of the pool inherit the monitor. If you want to exclude one or more pool members from inheriting the monitor that you assign to a pool, see *Excluding a pool member from a monitor*, on page 19-5.

To create a pool and assign a monitor

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **http_pool**.
4. For the **Health Monitors** setting, from the **Available** box select the name of the custom monitor, such as **my_http_monitor**, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. In the Resources section, ensure that the **Load Balancing Method** setting is set to **Round Robin.**
6. Ensure that the **Priority Group Activation** setting is set to **Disabled**.
7. For the **New Members** setting, add the pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IP address of a server in the pool.
   c) From the **Service Port** list, select **FTP**.
   d) Click **Add**.
   e) Repeat steps b, c, and d for each server in the pool.
8. Click **Finished**.

Excluding a pool member from a monitor

After you create the pool, you can exclude a pool member from inheriting a monitor in one of two ways:

- Removing the monitor from the pool member
- Assigning a different monitor to the pool member

Removing a monitor assignment from a pool member is useful if you want to monitor some, but not all, of the servers in a load balancing pool. When you exclude a pool member from inheriting the monitor that you assigned to the pool, you have the option of assigning a different monitor to that pool member. In this case, the other pool members are still monitored by the monitor you assigned to the pool itself.

To exclude a pool member from monitor you assigned to the pool, you must first follow the procedure described in *To create a pool and assign a monitor*, on page 19-4. Then you can use one of the following procedures.

To remove a monitor from a pool member

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   This displays a list of existing pools.
2. In the **Name** column, click the name of the pool you created in *Assigning a monitor to a pool*, on page 19-4.
3. On the menu bar, click **Members**.
   The Current Members area of the screen lists the members of the pool.
4. In the Members column, click the address of the pool member from which you want to remove the monitor.
   This displays the properties of that pool member.
5. From the **Configuration** list, select **Advanced**.
   This displays the **Health Monitors** setting.
6. From the **Health Monitors** list, select **None**.
7. Click **Update**.

To assign a unique monitor to a pool member

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   This displays a list of existing pools.
2. In the Name column, click the name of the pool you created in *Assigning a monitor to a pool*, on page 19-4.
3. On the menu bar, click **Members**.
   The Current Members area of the screen lists the members of the pool.

4. In the Members column, click the address of the pool member for which you want to assign a unique monitor.
   This displays the properties of that pool member.
5. From the **Configuration** list, select **Advanced**.
   This displays the **Health Monitors** setting.
6. From the **Health Monitors** list, select **Member Specific**.
7. Click **Update**.

Creating a virtual server

The last task in a basic configuration is to define a virtual server that references the pool that you created in *Creating a pool*, on page 19-4. You use the Configuration utility to create the virtual server.

To create a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_httppool**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. From the **Service Port** list, select a service.
6. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the pool you created (such as **http_pool**).
7. Click **Finished**.

20

Load Balancing Traffic to IPv6 Nodes

- Configuring the radvd service
- Configuring IPv4-to-IPv6 load balancing

Configuring the radvd service

The first task to setting up the BIG-IP system to function as an IPv4-to-IPv6 gateway is an optional one: to configure the **radvd** service. You configure the **radvd** service to send out ICMPv6 routing advisory messages, and to respond to ICMPv6 route solicitation messages.

When you perform this task, the BIG-IP system begins to support auto-configuration of downstream nodes. Also, the downstream nodes automatically discover that the BIG-IP system is their router.

Configuring the **radvd** service to perform these functions ultimately advertises the network's global address prefix on the internal VLAN. For more information on BIG-IP system services, see the ***Network and System Management Guide***.

To configure the radvd service

1. Using a serial console or the IP address of the BIG-IP system management interface, access a Linux prompt on the BIG-IP system.
2. Copy the file **/etc/radvd.conf.example** to a new file named **/etc/radvd.conf**.
3. Using the **nano** or **vi** text editor, open the file **/etc/radvd.conf**.
4. Using the example in the file, create an advertising configuration for the network's global address prefix.

   Note: *Replace the* ***prefix*** *option with an address appropriate for your network.*
5. Save the **/etc/radvd.conf** file and exit the editor.
6. Start the radvd service as follows:

   ```
   bigstart startup radvd
   ```
7. Verify that the IPv6 nodes have auto-configured their addresses for this prefix.
8. Take note of the addresses of the HTTP service IPv6 nodes. These addresses are required for the next step in the process, configuring IPv4-to-IPv6 load balancing.

Configuring IPv4-to-IPv6 load balancing

When you configure IPv4-to-IPv6 load balancing, you must create a pool for load balancing traffic to IPv6 nodes, and then create an IPv4 virtual server that processes application traffic.

Creating a pool of IPv6 nodes

The first task in configuring IPv4-to-IPv6 load balancing is to create a pool to load balance connections to IPv6 nodes. Use the Configuration utility to create this pool. For more detailed information about configuring a pool, see the ***Configuration Guide for Local Traffic Management***.

To create a pool of IPv6 nodes

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Pools**.
   The Pools screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Pool screen opens.
3. In the **Name** box, type a name for the pool, such as **ipv6_pool**.
4. For the **Health Monitors** setting, from the **Available** box select a monitor for the type of traffic you want to load balance, and click the Move button (**<<**) to move the monitor name to the **Active** box.
5. For the **New Members** setting, add the IPv6 pool members:
   a) Click the **New Address** option.
   b) In the **Address** box, type the IPv6 address of a node in the pool.
   c) In the **Service Port** box, type a service number, such as **80**, or select a service name, such as **HTTP**.
   d) Click **Add.**
   e) Repeat steps b, c, and d for each node in the pool.
6. Click **Finished**.

Creating a virtual server

The next task in a basic configuration is to define a virtual server that references the pool of IPv6 nodes. You use the Configuration utility to create the virtual server. For more information about configuring a virtual server, see the ***Configuration Guide for Local Traffic Management***.

To create a virtual server for IPv6 nodes

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the upper-right corner of the screen, click **Create**.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server, such as **vs_ipv6**.
4. In the **Destination** box, verify that the type of virtual server is **Host**, and in the **Address** box, type an IP address for the virtual server.
5. In the **Service Port** box, type a service number, such as **80**, or select a service name, such as **HTTP**, from the list.
6. In the Configuration area of the screen, locate the profile setting for the type of traffic you want to load balance, such as **HTTP Profile**, and select a profile name, such as **http**.
   This assigns the selected profile to the virtual server.
7. In the Resources area of the screen, locate the **Default Pool** setting and select the name of the pool you created in the previous section (for example, **ipv6_pool**).
8. Click **Finished**.

21

Mitigating Denial of Service and Other Attacks

- Basic denial of service security overview
- Configuring adaptive connection reaping
- Simple DoS prevention configuration
- Filtering out attacks with BIG-IP rules
- How the BIG-IP system handles several common attacks

Basic denial of service security overview

The BIG-IP system contains several features and configurations that provide you the ability to create a configuration that contributes to the security of your network. In particular, the BIG-IP system is in a unique position to mitigate some types of denial-of-service (DoS) attacks that try to consume system resources in order to deny service to the intended recipients.

The following features of the BIG-IP system help it resist many types of DoS attacks.

- **Hardened and dedicated kernel**
  The BIG-IP kernel has a mechanism built in to protect against SYN Flood attacks by limiting simultaneous connections, and tearing down connections that have unacknowledged SYN/ACK packets after some time period as passed. (A ***SYN/ACK*** packet is a packet that is sent as part of the TCP three-way handshake).
- **High performance**
  BIG-IP system can handle tens of thousands of Layer 4 (L4) connections per second. It would take a very determined attack to affect either the BIG-IP system itself, or the site, if sufficient server resources and bandwidth are available.
- **Large amount of available memory**
  SYN floods, or denial-of-service (DoS) attacks, can consume all available memory. The BIG-IP system supports a large amount of memory to help it resist DoS attacks.

This chapter describes several configurations that help mitigate DoS attacks. The configurations described include:

- How to configure the adaptive reapers to allow the BIG-IP system to respond to attacks, on page 21-1.
- A basic configuration to defend against denial of service attacks, on page 21-4.
- Several examples of rule syntax you can use to filter out specific known attacks, on page 21-7.

Configuring adaptive connection reaping

The BIG-IP system contains two global settings that provide the ability to reap connections adaptively. ***Connection reaping*** is when connections are removed from the BIG-IP system when the connection load uses enough memory to trigger the start of aggressive reaping. To prevent denial-of-service attacks, you can specify a low-water mark threshold and a high-water mark threshold:

- The ***low-water mark*** threshold determines at what point adaptive reaping becomes more aggressive.
- The ***high-water mark*** threshold determines when unestablished connections through the BIG-IP system will no longer be allowed. The value of this variable represents a percentage of memory utilization.

Once memory utilization has reached the high-water mark, connections are disallowed until the available memory has been reduced to the low-water mark threshold.

WARNING

The adaptive reaper settings do not apply to SSL connections. However, you can set TCP and UDP connection timeouts that reap idle SSL connections. For more information see ***Setting the TCP and UDP connection timers****, on page 21-4.*

To set the adaptive reapers using the Configuration utility

1. On the Main tab of the navigation pane, expand **System**, and click **General Properties**.
   The General screen opens.
2. From the Local Traffic menu, choose General.
   The System screen opens.
3. In the Properties table, set the following values:
   - Set the **Reaper High-water Mark** property to **95**.
   - Set the **Reaper Low-water Mark** property to **85**.
4. Click **Update**.

Tip

There is generally no need to change these values as they represent an optimal solution for most BIG-IP deployments.

Important

Setting both the adaptive reaper values to ***100*** *disables this feature.*

Logging adaptive reaper activity

You can log adaptive reaper activity by setting the logging levels on the BIG-IP system to a more sensitive level.

When you set this logging level, the system logs a rate-limited message (maximum once every 10 seconds), informing you that the adaptive reaping mode has been entered or exited. This log message has a priority of **warning**. For more information about the log level, refer to the **db** man page.

Important

*When the adaptive reaper high water limit is reached, the LCD displays the message **Blocking DoS Attack**.*

To set the adaptive reaper logging level from the command line

1. Open a console on the BIG-IP system.
2. Type the following command to view the adaptive reaper logging level:

   ```
   bp db Log.DosProtect.Level list
   ```

   The output looks like this:

   ```
   db Log.DosProtect.Level "Warning"
   ```

3. Choose the logging level for the adaptive reaper. The following levels display the message **Blocking DoS Attack** on the LCD when the **Reaper High Water Mark** is exceeded:

   - Emergency
   - Alert
   - Critical
   - Error
   - Warning

   The following levels do not display the **Blocking DoS Attack** message on the LCD.

   - Notice
   - Informational

4. Type the following command to set the adaptive reaper logging level, where **<log level>** is the logging level:

   ```
   bp db Log.DosProtect.Level "<log level>"
   ```

Simple DoS prevention configuration

DoS prevention is a simple configuration you can employ to mitigate the impact of denial-of-service attacks.

The configuration consists of four tasks:

- Set the global TCP and UDP connection reap times to 60 seconds.
- Set an IP rate class of 20Mbps, outstanding queue maximum size of 2Mbps.
- Set the connection limit on the main virtual server to the approximate amount of RAM in KB * 0.8.
- Set the global variable **Memory Restart Percent** to **97**.

Setting the TCP and UDP connection timers

You can set the TCP and UDP timers in the profile settings for the TCP profile and the UDP profiles. You should set these timers for the services that you use for your virtual servers. For example, **60** for HTTP connections, **60** for SSL connections.

To set the TCP connection reaper time

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Protocol menu, choose TCP.
   The TCP profile list screen opens.
3. Click the name of the profile you want to configure.
   The properties screen for the profile opens.
4. Set the **Idle Timeout** to **60**.
5. Click **Update**.

To set the UDP connection reaper time

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Protocol menu, choose UDP.
   The UDP profile list screen opens.
3. Click the name of the profile you want to configure.
   The properties screen for the profile opens.
4. Set the **Idle Timeout** to **60**.
5. Click **Update**.

Creating an IP rate class and applying it to a virtual server

The next task in setting up a simple configuration for DoS is to create a rate class. You must first create a rate class, and then apply the rate class to a virtual server.

Important

The rate class module requires a license key. If you do not have this functionality and you would like to purchase a license key, contact F5 Networks.

To create a rate class

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Rate Shaping**.
   The Rate Shaping screen displays.
2. Click the **Create** button to create a new rate class.
   The New Rate Class screen opens.
3. Configure the following class properties:
   - In the **Class Name** box, type the name you want to use for this class.
   - In the **Base Rate** box, type **2000000** (2 Mbps).
   - In the **Ceiling Rate** box, type **20000000** (20 Mbps).
   - In the **Burst Size** box, type **500**, and select **Megabytes** from the list.
   - From the **Direction** list, select **Any**.
   - From the **Queue Discipline** list, select **Stochastic Fair Queue**.
4. Click **Finished**.

After you create a rate class, you can apply it to the virtual servers in the configuration.

To apply a rate class to a virtual server

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers List screen opens.
2. On the Virtual Servers List screen, click the virtual you want to modify.
3. From the **Rate Class** list, select the rate class you created.
4. Click **Update**.

Setting connection limits on the main virtual server

This section describes how to set the connection limits on the main virtual server. The connection limits determine the maximum number of concurrent connections allowed on a virtual server. In this context, the main virtual server is the virtual server that receives the most traffic to your site.

To calculate a connection limit for the main virtual server

Before you set a connection limit, use the following formula to figure out what to set the connection limit value to on the main virtual server:

Connection Limit = Approximate Amount of RAM in KB * 0.8.

For example, if you have 256 MB of RAM, the calculation looks like this:

256,000 * 0.8 = 20480

In this case, you set the connection limit to **20480**.

To set the connection limits on the main virtual server using the Configuration utility

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers List screen opens.
2. On the Virtual Servers List screen, click the virtual server you want to modify.
3. In the **Connection Limit** box, type the number you calculated for the connection limit.
4. Click the **Update** button.

Setting the Memory Restart Percent

This section describes how to set the Memory Restart Percent. The value you type, **80** or higher, is the percentage of memory that is in use before the BIG-IP unit automatically reboots. This setting allows the BIG-IP unit to fail-over to a peer unit before all of the unit's memory is consumed.

To set the Memory Restart Percent

1. On the Main tab of the navigation pane, expand **System**, and click **General Properties**.
   The General screen opens.
2. In the Properties table, set the **Memory Restart Percent** to **97**.
3. Click **Update**.

Filtering out attacks with BIG-IP rules

You can create BIG-IP rules to filter out malicious DoS attacks. Once you identify a particular attack, you can write an iRule™ that discards packets that contain the elements that identify the packet as malicious.

◆ Note

For additional iRule syntax and examples, see the ***Configuration Guide for Local Traffic Management****.*

Filtering out a Code Red attack

The BIG-IP system is able to filter out the Code Red attack by using a rule to send the HTTP request to a dummy pool. For example, the following figure illustrates a rule that discards Code Red attacks.

```
when HTTP_REQUEST {
  if { [HTTP::uri] contains "default.ida" } {
     discard
  } else {
     pool RealServerPool
 }
}
```

Figure 21.1 *A sample rule for filtering out a Code Red attack*

Filtering out a Nimda attack

The Nimda worm is designed to attack systems and applications based on the Microsoft Windows operating system. For Nimda, a rule can be written as shown in Figure 21.2.

```
when HTTP_REQUEST {
   if { ([HTTP::uri] matches_regex ".*cmd.exe*.") or
([HTTP::uri] matches_regex ".*root.exe*.") or ([HTTP::uri]
matches_regex ".*admin.dll*.") } {
      discard
   } else {
      pool RealServerPool
 }
}
```

Figure 21.2 *Sample rule syntax to discard Nimda worm*

How the BIG-IP system handles several common attacks

You might want to know how the BIG-IP system reacts to certain common attacks that are designed to deny service by breaking the service or the network devices.

The following pages list the most common attacks, along with how the BIG-IP system functionality handles the attack.

WARNING

Take care any time you lower the idle session reaping time outs. It is possible that valid connections will be reaped if the application cannot respond in time.

SYN flood

A ***SYN flood*** is an attack against a system for the purpose of exhausting that system's resources. An attacker launching a SYN flood against a target system attempts to occupy all available resources used to establish TCP connections by sending multiple SYN segments containing incorrect IP addresses. Note that the term ***SYN*** refers to a type of connection state that occurs during establishment of a TCP/IP connection.

More specifically, a SYN flood is designed to fill up a SYN queue. A ***SYN queue*** is a set of connections stored in the connection table in the SYN-RECEIVED state, as part of the standard three-way TCP handshake. A SYN queue can hold a specified maximum number of connections in the SYN-RECEIVED state.

Connections in the SYN-RECEIVED state are considered to be half-open and waiting for an acknowledgement from the client. When a SYN flood causes the maximum number of allowed connections in the SYN-RECEIVED state to be reached, the SYN queue is said to be full, thus preventing the target system from establishing other legitimate connections. A full SYN queue therefore results in partially-open TCP connections to IP addresses that either do not exist or are unreachable. In these cases, the connections must reach their timeout before the server can continue fulfilling other requests.

Alleviating SYN flooding

The BIG-IP system includes a feature designed to alleviate SYN flooding. Known as ***SYN Check***, this feature sends information about the flow, in the form of cookies, to the requesting client, so that the system does not need to keep the SYN-RECEIVED state that is normally stored in the connection table for the initiated session. Because the SYN-RECEIVED state is not kept for a connection, the SYN queue cannot be exhausted, and normal TCP communication can continue.

The SYN Check feature complements the existing adaptive reaper feature in the BIG-IP system. While the adaptive reaper handles established connection flooding, SYN Check prevents connection flooding altogether. That is, while the adaptive reaper must work overtime to flush connections, the SYN Check feature prevents the SYN queue from becoming full, thus allowing the target system to continue to establish TCP connections.

You can configure the BIG-IP system to activate the SYN Check feature when some threshold of connections has been reached on one or all virtual servers. To set the threshold on an individual virtual server, you use the **bigpipe virtual** command. To set the threshold on all virtual servers, you use the **bigpipe global** command.

ICMP flood (Smurf)

The ***ICMP flood***, sometimes referred to as a "Smurf" attack, is an attack based on a method of making a remote network send ICMP Echo replies to a single host. In this attack, a single packet from the attacker goes to an unprotected network's broadcast address. Typically, this causes every machine on that network to answer with a packet sent to the target.

The BIG-IP system is hardened against these attacks because it answers only a limited number of ICMP requests per second, and then drops the rest.

On the network inside the BIG-IP system, the BIG-IP system ignores directed subnet broadcasts, and does not respond to the broadcast ICMP Echo that a Smurf attacker uses to initiate an attack.

You do not need to make any changes to the BIG-IP system configuration for this type of attack.

UDP flood

The ***UDP flood*** attack is most commonly a distributed denial-of-service attack (DDoS), where multiple remote systems are sending a large flood of UDP packets to the target.

The BIG-IP system handles these attacks similarly to the way it handles a SYN flood. If the port is not listening, the BIG-IP system drops the packets. If the port is listening, the reaper removes the false connections.

Setting the UDP idle session timeout to between 5 and 10 seconds reaps these connections quickly without impacting users with slow connections. However, with UDP this may still leave too many open connections, and your situation may require a setting of between 2 and 5 seconds.

UDP fragment

The ***UDP fragment*** attack is based on forcing the system to reassemble huge amounts of UDP data sent as fragmented packets. The goal of this attack is to consume system resources to the point where the system fails.

The BIG-IP system does not reassemble these packets, it sends them on to the server if they are for an open UDP service. If these packets are sent with the initial packet opening the connection correctly, then the connection is sent to the back-end server. If the initial packet is not the first packet of the stream, the entire stream is dropped.

You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Ping of Death

The ***Ping of Death*** attack is an attack with ICMP echo packets that are larger than 65535 bytes. Since this is the maximum allowed ICMP packet size, this can crash systems that attempt to reassemble the packet.

The BIG-IP system is hardened against this type of attack. However, if the attack is against a virtual server with the **Any IP** feature enabled, then these packets are sent on to the server. It is important that you apply the latest update patches to your servers.

You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Land attack

A ***Land*** attack is a SYN packet sent with the source address and port the same as the destination address and port.

The BIG-IP system is hardened to resist this attack. The BIG-IP system connection table matches existing connections so that a spoof of this sort is not passed on to the servers. Connections to the BIG-IP system are checked and dropped if spoofed in this manner.

You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Teardrop

A ***Teardrop*** attack is carried out by a program that sends IP fragments to a machine connected to the Internet or a network. The Teardrop attack exploits an overlapping IP fragment problem present in some common operating systems. The problem causes the TCP/IP fragmentation re-assembly code to improperly handle overlapping IP fragments.

The BIG-IP system handles these attacks by correctly checking frame alignment and discarding improperly aligned fragments.

You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Data attacks

The BIG-IP system can also offer protection from data attacks to the servers behind the BIG-IP system. The BIG-IP system acts as a port-deny device, preventing many common exploits by simply not passing the attack through to the server.

For information about rule examples for thwarting two common data attacks, see *Filtering out attacks with BIG-IP rules*, on page 21-7.

WinNuke

The ***WinNuke*** attack exploits the way certain common operating systems handle data sent to the NetBIOS ports. NetBIOS ports are **135**, **136**, **137** and **138**, using TCP or UDP. The BIG-IP system denies these ports by default.

On the BIG-IP system, do not open these ports unless you are sure your servers have been patched against this attack.

Sub 7

The ***Sub 7*** attack is a Trojan horse that is designed to run on certain common operating systems. This Trojan horse allows the system to be controlled remotely.

This Trojan horse listens on port **27374** by default. The BIG-IP system does not allow connections to this port from the outside, so a compromised server cannot be controlled remotely.

Do not open high ports (ports above **1024**) without explicit knowledge of what applications will be running on these ports.

Back Orifice

Back Orifice is a Trojan horse that is designed to run on certain common operating systems. This Trojan horse allows the system to be controlled remotely.

This Trojan horse listens on UDP port **31337** by default. The BIG-IP system does not allow connections to this port from the outside, so a compromised server cannot be controlled remotely. Do not open high ports (ports above **1024**) without explicit knowledge of what will be running on these ports.

22

Configuring Remote Authentication for Management Traffic

- Introducing remote authentication for BIG-IP system management traffic
- Configuring LDAP- or Active Directory-based authentication
- Configuring RADIUS-based authentication

Introducing remote authentication for BIG-IP system management traffic

As an administrator in a large computing environment, you might prefer to store user accounts remotely, on a dedicated authentication server. Using a remote authentication server, the BIG-IP system can authenticate two types of network traffic:

- **Application traffic that is slated for load balancing**
  This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. To configure remote authentication for this type of traffic, see Chapter 23, *Configuring Remote Authentication for Application Traffic*, and the ***Configuration Guide for Local Traffic Management***.
- **Management traffic for administering the BIG-IP system**
  This type of traffic does not pass through a virtual server, and instead passes through the management interface (MGMT). You configure remote authentication for this type of traffic when you create your administrative user accounts, storing them on a remote authentication server. ***Administrative user accounts*** are accounts that you create for the system and network administrators who manage the BIG-IP system. For more information, see the remainder of this chapter and also the ***BIG-IP Network and System Management Guide***.

When you want to use a remote server to authenticate traffic that manages the BIG-IP system, you can store BIG-IP system administrative accounts on one of three authentication server types:

- A Lightweight Directory Access Protocol (LDAP) server
- A Microsoft® Windows Active Directory server
- A Remote Authentication Dial-in User Service (RADIUS) server

By default, the BIG-IP system uses basic HTTP authentication (using a user name and password) when remotely authenticating management traffic.

The procedure you use to set up remote authentication of management traffic depends on which type of remote server you are using to store the user accounts.

Configuring LDAP- or Active Directory-based authentication

You can configure the BIG-IP system to use an LDAP or Microsoft® Windows Active Directory server for authenticating BIG-IP system management traffic, that is, traffic that passes through the management interface (MGMT). By default, user credentials are based on basic HTTP authentication (that is, user name and password).

If the remote authentication server is set up to authenticate SSL traffic, there is an additional feature that you can enable. You can configure the BIG-IP system to perform the server-side SSL handshake that the remote server would normally perform when authenticating client traffic. In this case, there are some preliminary steps you must perform to prepare for remote authentication using SSL.

To prepare for SSL-based remote authentication

1. Convert the Certificate Authority (CA) or self-signed certificates to PEM format.
2. On the BIG-IP system, import the certificates, using the Configuration utility.
   You can store the certificates in any location on the BIG-IP system. For information on importing certificates, see the ***BIG-IP Network and System Management Guide***.

Once you have performed these preliminary SSL tasks, you can enable SSL as part of the procedure described in *To configure remote LDAP- or Active Directory-based authentication*, following.

To configure remote LDAP- or Active Directory-based authentication

1. On the Main tab of the navigation pane, expand **System**, and click **Users**.
   The Users screen opens.
2. On the menu bar, click **Authentication Source**.
   The Authentication Source screen opens.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - Active Directory** or **Remote - LDAP**.
5. In the **Host** box, type the IP address of the remote server.
6. For the **Port** setting, retain the default port number (**389**) or type a new port number in the box.
   This setting represents the port number that the BIG-IP system uses to access the remote server.

7. In the **Remote Directory Tree** box, type the file location (tree) of the user authentication database on the LDAP or Active Directory server. At minimum, you must specify a domain component (that is, **dc=<value>**).
8. For the **Scope** setting, retain the default value (**Sub**) or select a new value.
   This setting specifies the level of the remote server database that the BIG-IP system should search for user authentication. For more information on this setting, see the online help.
9. For the **Bind** setting, specify a user ID login for the remote server:
   a) In the **DN** box, type the Distinguished Name for the remote user ID.
   b) In the **Password** box, type the password for the remote user ID.
   c) In the **Confirm** box, re-type the password that you typed in the Password box.
10. If you want to enable SSL-based authentication, click the **SSL** box and, if necessary, configure the following settings.

    Important: *Be sure to specify the full path name of the storage location on the BIG-IP system. For example, if the certificate is stored in the directory* ***/config/bigconfig/ssl.crt****, type the value* ***/config/bigconfig/ssl.crt****.*

    a) In the **SSL CA Certificate box**, type the name of a chain certificate, that is, the third-party CA or self-signed certificate that normally resides on the remote authentication server.
    b) In the **SSL Client Key** box, type the name of the client SSL key. Use this setting only in the case where the remote server requires that the client present a certificate. If a client certificate is not required, you do not need to configure this setting.
    c) In the **SSL Client Certificate** box, type the name of the client SSL certificate.
    Use this setting only in the case where the remote server requires that the client present a certificate. If a client certificate is not required, you do not need to configure this setting.
11. Click **Finished**.

Configuring RADIUS-based authentication

You can configure the BIG-IP system to use a RADIUS server for authenticating BIG-IP system management traffic, that is, traffic that passes through the management interface (MGMT). By default, user credentials are based on basic HTTP authentication (that is, user name and password).

To configure remote RADIUS-based authentication

1. On the Main tab of the navigation pane, expand **System**, and click **Users**.
   The Users screen opens.
2. On the menu bar, click **Authentication Source**.
   The Authentication Source screen opens.
3. Click **Change**.
4. From the **User Directory** list, select **Remote - RADIUS**.
5. For the **Primary** setting, configure these settings:
   a) In the **Host** box, type the IP address of the remote server.
   b) In the **Port** box, retain the default port number (**1812**) or type a new port number in the box.
   This setting represents the port number that the BIG-IP system uses to access the remote server.
   c) In the **Secret** box, type the RADIUS secret.
   d) In the **Confirm** box, re-type the secret that you typed in the **Secret** box.
   Note that the values of the **Secret** and **Confirm** settings must match.
6. If you want to configure a secondary RADIUS server in the event that the primary server becomes unavailable, locate the **Secondary** setting and check the **Configure Secondary Host** box.
   This causes additional settings to appear.
7. Configure the remaining settings for the secondary server, using the instructions for the primary server in step 5.
8. Click **Finished**.

23

Configuring Remote Authentication for Application Traffic

- Introducing remote authentication for application traffic
- Configuring authentication that uses a remote LDAP or Active Directory server
- Configuring authentication that uses a remote RADIUS server
- Configuring authentication that uses a remote TACACS+ server

Introducing remote authentication for application traffic

As an administrator in a large computing environment, you might prefer to store your site's user accounts remotely, on a dedicated authentication server. Fortunately, you can set up the BIG-IP system to use this server to authenticate any network traffic passing through the BIG-IP system. Remote authentication servers typically use these protocols: Lightweight Directory Access Protocol (LDAP), Microsoft® Windows Active Directory, Remote Authentication Dial-in User Service (RADIUS), and TACACS+, derived from Terminal Access Controller Access Control System (TACACS).

Using a remote authentication server, the BIG-IP system can authenticate two types of network traffic:

- **Application traffic that is slated for load balancing**
  This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. For more information, see the remainder of this chapter, and the ***Configuration Guide for Local Traffic Management***.
- **Management traffic for administering the BIG-IP system**
  This type of traffic does not pass through a virtual server, and instead passes through the management interface (MGMT). You configure remote authentication for this type of traffic when you create your administrative user accounts. ***Administrative user accounts*** are accounts that you create for the system and network administrators who manage the BIG-IP system. For more information, see Chapter 22, *Configuring Remote Authentication for Management Traffic* in this guide, and the ***BIG-IP Network and System Management Guide***.

When you want to use a remote server to authenticate application traffic passing through the BIG-IP system, you can use one of these server types:

- An LDAP or Active Directory server
- A RADIUS server
- A TACACS+ server

To configure remote user authentication for application traffic, you must create both a configuration object and an authentication profile. Each authentication server type requires a different configuration object and profile. For example, to configure the BIG-IP system to use an LDAP authentication server, you must create an LDAP configuration object and a custom LDAP profile. When implementing RADIUS authentication, you must also create a third type of object, called a RADIUS server object.

Configuring authentication that uses a remote LDAP or Active Directory server

You can configure the BIG-IP system to use an LDAP or Active Directory server for authenticating traffic that passes through the TMM interfaces of the BIG-IP system. By default, client credentials are based on basic HTTP authentication (that is, user name and password). However, you can also enable SSL authentication, which is based on SSL keys and certificates.

To configure LDAP or Active Directory authentication for application traffic, you complete these tasks:

- Create an LDAP-type configuration object
- Create an LDAP-type authentication profile
- Modify a virtual server that is configured to manage HTTP traffic

Creating an LDAP configuration object

The first task in configuring LDAP-based or Active Directory-based remote authentication on the BIG-IP system is to create a custom LDAP configuration object, using the Configuration utility. An ***LDAP configuration object*** specifies information that the BIG-IP system needs to perform the remote authentication. For example, the configuration object specifies the remote LDAP tree that the system uses as the source location for the authentication data.

If the remote authentication server uses LDAP or Active Directory and is set up to authenticate SSL authentication traffic, there is an additional feature that you can enable. You can configure the BIG-IP system to perform the server-side SSL handshake that the remote server would normally perform when authenticating client traffic. In this case, there are some preliminary tasks you must perform to prepare for remote authentication using SSL.

To prepare for SSL-based remote authentication

1. Convert the Certificate Authority (CA) or self-signed certificates to PEM format.
2. On the BIG-IP system, import the certificates, using the Configuration utility.
   You can store the certificates in any location on the BIG-IP system. For information on importing certificates, see the ***Configuration Guide for Local Traffic Management***.

Once you have performed these preliminary SSL tasks, you can enable SSL-based remote server authentication. You do this as part of creating the LDAP configuration object, which includes these Advanced settings:

- **SSL CA Certificate**
  This represents the name of the certificate that normally resides on the remote authentication server.
- **SSL Client Key**
  This represents the name of the SSL key that the client sends to the BIG-IP system. This key specification is only necessary when the remote server requires a client certificate.
- **SSL Client Certificate**
  This represents the name of the SSL certificate that the client sends to the BIG-IP system. This certificate specification is only necessary when the remote server requires a client certificate.

Important

When specifying key and certificate files while creating an LDAP configuration object, be sure to specify the full path name of the storage location on the BIG-IP system. For example, if the certificate is stored in the directory ***/config/bigconfig/ssl.crt****, type the value* ***/config/bigconfig/ssl.crt****.*

After you create the custom LDAP configuration object, you create a custom LDAP profile, and then assign the custom profile to an HTTP virtual server.

To create a custom LDAP configuration object

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Authentication menu, choose Configurations.
   The Authentication Configurations screen opens.
3. On the upper-right corner of the screen, click **Create**.
   The New Authentication Configuration screen opens.
4. In the **Name** box, type a unique name for the configuration object, such as **my_ldap_config**.
5. From the **Type** list, select **LDAP**.
   This displays the configuration object settings that you can configure.
6. For the Configuration area, select **Basic** or **Advanced**.
   Selecting **Advanced** causes additional settings to appear on the screen.
7. In the **Remote LDAP Tree** box, type the file location (tree) of the user authentication database on the LDAP or Active Directory server.
   At a minimum, you must specify a domain component (that is, **dc=<value>**).
8. For the **Hosts** setting:
   a) Type the IP address of the remote LDAP or Active Directory server.
   b) Click **Add**.
      The IP address appears in the text window.
9. Retain or change the **Service Port** value.
10. Retain or change the **LDAP Version** value.
11. If you selected a basic configuration in step 6, click **Finished**. If you selected an advanced configuration in step 6, configure the remaining settings and click **Finished**.
    For descriptions of all advanced settings, see the online help or the ***Configuration Guide for Local Traffic Management***.

◆ Note

For information about enabling SSL authentication, see the beginning of this section, ***Creating an LDAP configuration object****, on page 23-2.*

Creating an LDAP authentication profile

The next task in configuring LDAP-based or Active Directory-based remote authentication on the BIG-IP system is to create a custom LDAP profile. An ***LDAP profile*** specifies information such as the LDAP authentication mode (**Enabled** or **Disabled**), and the name of the LDAP configuration object you previously created.

After you create the custom LDAP profile, you assign the custom profile and a default iRule to a virtual server.

To create a custom LDAP profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Authentication menu, choose Profiles.
   The Authentication Profiles screen opens.
3. On the upper-right corner of the screen, click **Create**.
   The New Authentication Profile screen opens.
4. In the **Name** box, type a unique name for the profile, such as **my_ldap_profile**.
5. From the **Type** list, select **LDAP**.
   This displays the profile settings that you can configure.
6. From the **Parent Profile** list, verify that **ldap** is selected
   This causes the new profile to inherit its default configuration values from the default profile, named **ldap**.
7. From the **Configuration** list, select the name of the LDAP configuration object that you previously created.
8. For all remaining settings, retain the default values.
9. Click **Finished**.

Modifying a virtual server for LDAP authentication

The final task in the process of implementing authentication using a remote LDAP server is to assign the custom LDAP profile and a default LDAP authentication iRule to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

◆ Note

The virtual server to which you assign the profiles and the iRule must be a Standard type of virtual server.

To modify a virtual server for LDAP authentication

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the Name column, click the name of a Standard-type virtual server to which an HTTP profile is assigned.
   This displays the properties of that virtual server.
3. From the **Configuration** list, select **Advanced**.
   This displays additional properties.
4. From the **Authentication Profiles** list, from the **Available** box select the name of the custom LDAP profile that you previously created, and click the Move button (**<<**).
   This moves the profile name to the **Enabled** box.
5. Click **Update**.

Configuring authentication that uses a remote RADIUS server

A RADIUS authentication module is a mechanism for authenticating client connections passing through a BIG-IP system. You use this module when your authentication data is stored on a remote RADIUS server. In this case, client credentials are based on basic HTTP authentication (that is, user name and password).

To implement a RADIUS authentication module, you must configure the BIG-IP system to access data on a remote RADIUS server. To do this, you must create:

- One or more high-level RADIUS server objects
- A RADIUS configuration object
- A RADIUS profile
- Modify a virtual server to assign the RADIUS profile to it.

Creating a RADIUS server object

The next task in configuring RADIUS-based remote authentication on the BIG-IP system is to create a custom RADIUS configuration object. A ***RADIUS configuration object*** specifies information that the BIG-IP system needs to perform the remote authentication.

After you create the custom RADIUS server object, you create a custom RADIUS configuration object and a custom RADIUS profile, and then assign the custom profile and a default iRule to an HTTP virtual server.

To create a RADIUS server object

1. On the Main tab, expand **Local Traffic**.
2. Click **Profiles**.
   The Profiles screen opens.
3. From the Authentication menu, choose RADIUS Servers.
   This displays the RADIUS Server List screen.
4. In the upper right corner of the screen, click **Create**.
5. For the **Name** setting, type a unique name for the RADIUS server object, such as **my_radius_server**.
6. For the **Server** setting, type a host name or IP address for the remote RADIUS server.
7. For the **Secret** and **Confirm Secret** settings, type the RADIUS secret.

8. Retain the default **Timeout** value.
9. Click **Finished**.

Creating a RADIUS configuration object

The next task in configuring RADIUS-based remote authentication on the BIG-IP system is to create a custom RADIUS configuration object. A ***RADIUS configuration object*** specifies information that the BIG-IP system needs to perform the remote authentication.

After you create the custom RADIUS configuration object, you create a custom RADIUS profile, and then assign the custom profile and a default iRule to an HTTP virtual server.

To create a RADIUS configuration object

1. On the Main tab, expand **Local Traffic**, and click **Profiles**.
   The Profiles screen opens.
2. From the Authentication menu, choose Configurations.
3. In the upper right corner of the screen, click **Create**.
   This displays the New Configuration screen.
4. For the **Name** setting, specify a unique name for the configuration object, such as **my_radius_config**.
5. For the **Type** setting, select **RADIUS**.
   The screen expands to show several settings.
6. From the **Configuration** list, select **Basic** or **Advanced**.
   Selecting **Advanced** causes additional settings to appear on the screen.
7. For the **RADIUS Servers** setting, from the **Available** box select the IP address of the RADIUS server and click the Move button (**<<**).
   This moves the server name to the **Selected** box.
8. In the **Client ID** box, type a **NAS-Identifier** string.
   Required for RADIUS authentication, the **NAS-Identifier** string appears in Access-Request packets and identifies the NAS that originates the packet. An example of a **NAS-Identifier** string is a fully-qualified domain name (FQDN).
9. If you selected a basic configuration in step 7, click **Finished**. If you selected an advanced configuration in step 7, configure the remaining settings and click **Finished**.
   For descriptions of all advanced settings, see the online help or the ***Configuration Guide for Local Traffic Management***.

Creating a RADIUS profile

The next task in configuring RADIUS-based remote authentication on the BIG-IP system is to create a custom RADIUS profile. A ***RADIUS profile*** specifies information such as the RADIUS authentication mode (**Enabled** or **Disabled**), and the name of the RADIUS configuration object you previously created.

After you create the profile, you assign the custom profile and a default iRule to an HTTP virtual server.

To create a custom RADUS profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Authentication menu, choose Profiles.
   The Authentication Profiles screen opens.
3. On the upper-right corner of the screen, click **Create**.
   The New Authentication Profile screen opens.
4. From the **Type** list, select **RADIUS**.
   This displays the profile settings that you can configure.
5. In the **Name** box, type a unique name for the profile, such as **my_radius_profile**.
6. From the **Parent Profile** list, verify that **radius** is selected.
   This causes the new profile to inherit its default configuration values from the default profile, named **ldap**.
7. From the **Configuration** list, select the name of the RADIUS configuration object that you previously created.
8. For all remaining settings, retain the default values.
9. Click **Finished**.

Modifying a virtual server for RADIUS authentication

The final task in the process of implementing authentication using a remote RADIUS server is to assign the custom RADIUS profile to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

◆ Note

The virtual server to which you assign an authentication profile must be a Standard type of virtual server.

To modify a virtual server for RADIUS authentication

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the Name column, click the name of a virtual server.
   This displays the properties of that virtual server.
3. From the **Configuration** list, select **Advanced**.
   This displays additional properties.
4. From the **Authentication Profiles** list, from the **Available** box select the name of the custom RADIUS profile that you previously created, and click the Move button (**<<**).
   This moves the profile name to the **Enabled** box.
5. Click **Update**.

Configuring authentication that uses a remote TACACS+ server

You can configure the BIG-IP system to use a TACACS+ server for authenticating traffic that passes through the TMM interfaces of the BIG-IP system. In this case, client credentials are based on basic HTTP authentication (that is, user name and password).

To configure LDAP or Active Directory authentication for application traffic, you complete these tasks:

- Create an LDAP-type configuration object
- Create an LDAP-type authentication profile
- Modify a virtual server configured to manage HTTP traffic

Creating a TACACS+ configuration object

The first task in configuring TACACS+ remote authentication on the BIG-IP system is to create a custom TACACS+ configuration object. A TACACS+ ***configuration object*** specifies information that the BIG-IP system needs to perform the remote authentication. For example, the configuration object specifies the IP address of the remote TACACS+ server.

To create a custom TACACS+ configuration object

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Authentication menu, choose Configurations.
   The Authentication Configurations screen opens.
3. On the upper-right corner of the screen, click **Create**.
   The New Authentication Configuration screen opens.
4. From the **Type** list, select **TACACS+**.
   This displays the configuration object settings that you can configure.
5. In the **Name** box, type a unique name for the configuration object, such as **my_tacacs_config**.
6. For the Configuration area, select **Basic** or **Advanced**.
   Selecting **Advanced** causes additional settings to appear on the screen.
7. In the **Servers** box, type the IP address of the remote TACACS+ server and click **Add.**
   The IP address appears in the text box.

8. For the **Hosts** setting, type the IP address of the remote LDAP or Active Directory server and click **Add**.
   The IP address appears in the text window.
9. In the **Secret** box, type a TACACS+ secret. key to be used for encrypting or decrypting packets sent to or from the server.
10. In the **Confirm Secret** box, re-type the secret key you typed in the **Secret** box.
11. If you selected a basic configuration in step 6, click **Finished**. If you selected an advanced configuration in step 6, configure the remaining settings and click **Finished**.
    For descriptions of all advanced settings, see the online help or the ***Configuration Guide for Local Traffic Management***.

Once you have created the TACACS+ configuration object, you must create a custom TACACS+ profile and modify an HTTP virtual server.

Creating a TACACS+ profile

The next task in configuring TACACS+-based remote authentication on the BIG-IP system is to create a custom TACACS+ profile. After you create the profile, you assign the custom profile, the default **http** profile, and a default iRule to a virtual server.

To create a custom TACACS+ profile

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Profiles**.
   The HTTP Profiles screen opens.
2. From the Authentication menu, choose Profiles.
   The Authentication Profiles screen opens.
3. On the upper-right corner of the screen, click **Create**.
   The New Authentication Profile screen opens.
4. From the **Type** list, select **TACACS+**.
   This displays the profile settings that you can configure.
5. In the **Name** box, type a unique name for the profile, such as **my_tacacs_profile**.
6. From the **Parent Profile** list, verify that **tacacs** is selected.
   This causes the new profile to inherit its default configuration values from the default profile, named **tacacs**.
7. From the **Configuration** list, select the name of the TACACS+ configuration object that you previously created.
8. For all remaining settings, retain the default values.
9. Click **Finished**.

Modifying a virtual server for TACACS+ authentication

The final task in the process of implementing authentication using a remote TACACS+ server is to assign the custom TACACS+ profile and an existing default authentication iRule to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

Note

The virtual server to which you assign an authentication profile and iRule must be a Standard type of virtual server.

To modify a virtual server for TACACS+ authentication

1. On the Main tab of the navigation pane, expand **Local Traffic**, and click **Virtual Servers**.
   The Virtual Servers screen opens.
2. In the Name column, click the name of a virtual server.
   This displays the properties of that virtual server.
3. From the **Configuration** list, select **Advanced**.
   This displays additional properties.
4. From the **Authentication Profiles** list, from the **Available** box select the name of the custom TACACS+ profile that you previously created, and click the Move button (**<<**).
   This moves the profile name to the **Enabled** box.
5. Click **Update**.

Index

G

H

I

K

L

M

N

T

U

V

W