

## Services

### Solutions in this chapter:

- DHCP Functionality
  - PPPoE
  - EasyVPN
  - Routing and the PIX Firewall
  - Queuing and Policing
- 
- ☑ Summary
  - ☑ Solutions Fast Track
  - ☑ Frequently Asked Questions

# Introduction

In addition to performing its traditional firewall functions such as filtering traffic, the PIX firewall can also provide a variety of other services. These services are a convenient way to get added value from your firewall; rather than having to set up separate servers and applications to deliver these services to your network, the firewall becomes an all-in-one appliance.

## DHCP Functionality

DHCP is a convenient method of providing required configuration parameters to network nodes, such as IP address, default gateway, DNS servers, and WINS servers. Rather than configuring these parameters manually on every client, DHCP allows the configuration details to be set centrally, in this case on the PIX firewall, and then assigned to each node as required.

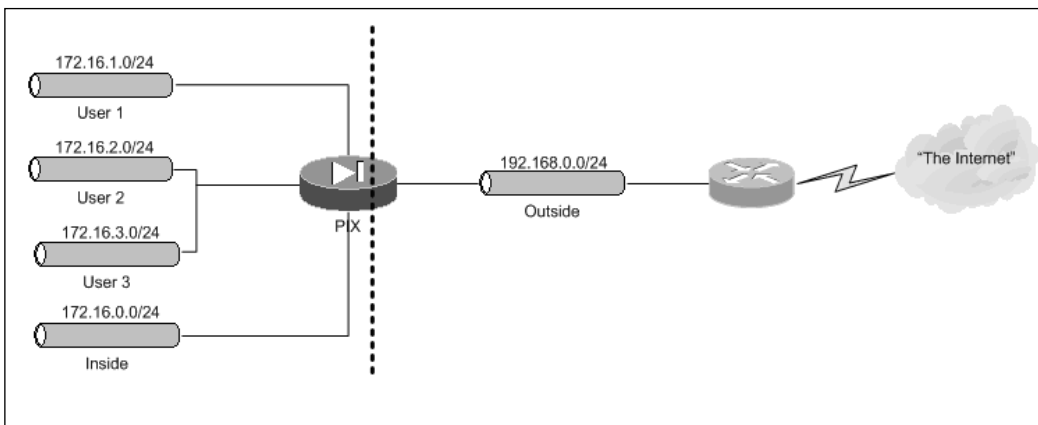
The PIX firewall is capable of acting as a DHCP server to a node connected to any of its interfaces. The firewall is also capable of acting as a DHCP relay server, where it forwards DHCP requests from clients to another DHCP server. Finally, the firewall has DHCP client functionality, allowing for the configuration of its own network parameters based on another DHCP server on the network.

## DHCP Servers

You can configure the PIX firewall to issue IP addresses, as well as information such as DNS and WINS servers, the default gateway, and a DNS domain name. The process for configuring DHCP is relatively straightforward: define your parameters associated with an interface. *Do not forget to enable DHCPD on the appropriate interface!*

In Figure 7.1, we have four networks for which we want the PIX firewall to issue IP addresses.

**Figure 7.1** PIX DHCP Services



Notice how we have defined four separate pools and setups of parameters. These serve to illustrate that the PIX can uniquely provide this service to differing networks. The configuration for providing DHCP services to each of these networks is provided.

```
dhcpcd address 172.16.0.100-172.16.0.200 inside
dhcpcd dns 192.168.0.200
dhcpcd wins 192.168.0.100
dhcpcd lease 6000
dhcpcd domain inside.syngress.com
dhcpcd enable inside
```

```
dhcpcd address 172.16.1.100-172.16.1.200 User1
dhcpcd dns 192.168.0.200
dhcpcd wins 192.168.0.100
dhcpcd lease 6000
dhcpcd domain user1.syngress.com
dhcpcd enable user1
```

```
dhcpcd address 172.16.2.100-172.16.2.200 User2
dhcpcd dns 192.168.0.200
dhcpcd wins 192.168.0.100
dhcpcd lease 6000
dhcpcd domain user2.syngress.com
dhcpcd enable user2
```

```
dhcpcd address 172.16.3.100-172.16.3.200 User3
dhcpcd dns 192.168.0.200
dhcpcd wins 192.168.0.100
dhcpcd lease 6000
dhcpcd domain user3.syngress.com
dhcpcd enable user3
```

Commands are available for checking the state of the server. For example:

```
PIX1(config)# show dhcpcd
dhcpcd address 192.168.2.201-192.168.2.210 inside
dhcpcd lease 3000
dhcpcd ping_timeout 750
dhcpcd dns 1.2.3.4 1.2.3.31
dhcpcd enable inside
```

Other commands show the current state of IP bindings (which client has been assigned which IP address) and general server statistics:

```
PIX1(config)# show dhcpd binding
IP Address Hardware Address Lease Expiration Type
192.168.2.210 0100.a0c9.777e 84985 seconds automatic
```

Here, a client with MAC address 0100.a0c9.777e has obtained IP address 192.168.2.210, and this lease will expire in 84,985 seconds:

```
PIX1(config)# show dhcpd statistics
Address Pools 1
Automatic Bindings 1
Expired Bindings 1
Malformed messages 0
Message Received
BOOTREQUEST 0
DHCPDISCOVER 1
DHCPREQUEST 2
DHCPDECLINE 0
DHCPRELEASE 0
DHCPINFORM 0
Message Sent
BOOTREPLY 0
DHCPOFFER 1
DHCPACK 1
DHCPNAK 1
```

These statistics show the number of IP address pools configured, the number of active leases (bindings), expired bindings, messages received with errors, and a detailed breakdown on message type for correctly received and sent messages.

## Cisco IP Phone-Related Options

In addition to the standard DHCP parameters such as IP address and default gateway, a Cisco IP phone makes use of DHCP to obtain the IP address of the TFTP server from which the phone can download its configuration. To configure the PIX firewall to respond to IP phone DHCP requests, enter the following commands:

```
PIX1(config)# dhcpd option 66 ascii 172.16.0.5
PIX1(config)# dhcpd option 3 ip 172.16.0.1
```

The first command will assign the IP phone a TFTP server of 172.16.0.5, and the second command will assign the phone a default gateway of 172.16.0.1. Note that you have the option of assigning more than one TFTP server. In this case, enter the following commands:

```
PIX1(config)# dhcpd option 150 ascii 172.16.0.5 172.16.0.6
```

This command assigns the phone 172.16.0.5 as the primary TFTP server and 172.16.0.6 as the secondary TFTP server.

## DHCP Relay

If you want to use the PIX as a DHCP server, you can configure it to relay requests to a DHCP server located elsewhere. This is similar to the IP helper command in IOS. You simply specify the target DHCP server address and the interface to which it is associated. For example, the following command is used to relay DHCP requests to 192.168.0.250:

```
dhcprelay server 192.168.0.250 outside
```

## DHCP Clients

When configured as a DHCP client, the PIX firewall can obtain the configuration of its outside interface from a designated DHCP server—for example, a server located at an ISP. This configuration includes the IP address, the subnet mask, and optionally, the default route.

### NOTE

---

The DHCP client feature can only be configured on the “outside” interface of the PIX firewall.

---

This address can be used, for example, as a PAT address for all outgoing communications. This is configured in the following way (assuming that the DHCP client is already configured):

```
nat (inside) 1 0 0
global (outside) 1 interface
```

This configuration will work with any IP address assigned to the outside interface by DHCP. The configuration of the DHCP client is rather simple, and all you need to use is the following command:

```
ip address outside dhcp [setroute] [retry <retry_cnt>]
```

You do this instead of specifying a fixed IP address for an outside interface. The optional *setroute* keyword forces the PIX firewall to pick up the IP address, the subnet mask, and the default route. Do not configure a static default route on the firewall if you use the *setroute* option. The *retry* option tells the PIX firewall to try to contact a DHCP server a specified number of times before giving up. If this keyword is not specified, no retries are attempted. If this keyword is specified but no retry count is given, the default number of retries is four. For example, the following command configures a DHCP client on the outside interface to

obtain an IP address, subnet mask, and default route from the DHCP server, and only one attempt will be made:

```
PIX1 (config) # ip address outside dhcp setroute
```

The following command configures the DHCP client to obtain an IP address and subnet mask only and tries at least five times before giving up if no DHCP servers are available:

```
PIX1 (config) # ip address outside dhcp retry 5
```

There are no special commands for renewing and releasing a DHCP lease; simply issue the same command again and the lease will be renewed. The address obtained can be viewed using:

```
PIX1# show ip address outside dhcp
```

This produces output similar to the following:

```
Temp IP Addr:123.1.2.3 for peer on interface:outside
Temp sub net mask:255.255.255.0
DHCP Lease server:123.1.2.31, state:3 Bound
DHCP Transaction id:0x4567
Lease:259200 secs, Renewal:129600 secs, Rebind:226800 secs
Temp default-gateway addr:123.1.2.1
Next timer fires after:100432 secs
Retry count:0, Client-ID:cisco-0000.0000.0000-outside
```

This output means that PIX has obtained an IP address of 123.1.2.3 and a subnet mask of 255.255.255.0 from the DHCP server 123.1.2.31. This DHCP lease is granted for 259200 seconds with renewal time of 129600 seconds. Time left until the next renewal is 100432 seconds, and there were no retries in contacting the server.

In case there are any issues with the DHCP client, you can troubleshoot using *debug* commands, including *debug dhcp packet*, *debug dhcp detail*, and *debug dhcp error*. The commands are self-explanatory. *debug dhcp packet* displays all DHCP traffic between the PIX client and a remote server, the *detail* option shows details of negotiation, and the *error* option displays all errors in this communication.

## PPPoE

Point-to-Point Protocol over Ethernet (PPPoE), documented in RFC 2516, is an encapsulation of Point-to-Point Protocol (PPP, RFC 1661) for Ethernet networks (which include DSL modems and cable connections). PPPoE is often used in SOHO environments because it allows ISPs to use their existing remote access infrastructure and, as its most important feature, allows authenticated IP address assignment. PPPoE links are established in two main phases:

- **Active discovery phase** During this first phase, a PPPoE client attempts discovery of the PPPoE server, also called the *address concentrator* (AC). The PPPoE layer is established and a session ID is assigned.

- **PPP session phase** A PPP link is established (encapsulated in Ethernet) by the usual means: options and link layer protocols are negotiated, etc. PPP authentication (PAP, CHAP, or MS-CHAP) is performed.

## NOTE

---

PPPoE is not currently supported by version 7.

---

After the session is established, data travels between endpoints encapsulated in PPPoE headers.

The PIX firewall supports PPPoE since software v6.2. Most of the PPPoE configuration is performed using the *vpdn* command. PPPoE configuration starts with configuring the username and password to be used by the PIX in establishing a link to the server.

## NOTE

---

The PIX only supports PPPoE client functionality. PPPoE clients can be enabled only on the outside interface at this time (v6.2).

---

First, a VPDN group needs to be created:

```
vpdn group <group_name> request dialout pppoe
```

The *group\_name* parameter can be anything you like. It is used to group all PPPoE settings together. For example:

```
PIX1 (config) # vpdn group my-pppoe-group request dialout pppoe
```

Then, the authentication type needs to be selected (if required by an ISP):

```
vpdn group <group_name ppp> authentication pap | chap | mschap
```

PAP is Password Authentication Protocol, CHAP is Challenge-Handshake Authentication Protocol, and MS-CHAP is Microsoft's version of CHAP. With the same group name, this command selects an authentication protocol for this specific PPPoE group—for example, with CHAP authentication:

```
PIX1 (config) # vpdn group my-pppoe-group ppp authentication chap
```

Your ISP assigns the username and password to your system, and they are configured on PIX with the following commands:

```
vpdn group <group_name> localname <username>
vpdn username <username> password <pass>
```

The second of these commands associates a username with the password, and the first command assigns the username to be used for a specific group; for example:

```
PIX1 (config) # vpdn group my-ppoe-group localname witt
PIX1 (config) # vpdn username witt password cruelmail
```

These commands assign the username *witt* and password *cruelmail* to be used for the PPPoE dial-out group *my-ppoe-group*. After configuring authentication, the next task is to enable the PPPoE client on the PIX. This is done in the configuration of the outside interface with the *ip address outside pppoe [setroute]* command. After this command is entered, the current PPPoE session is terminated and a new one is established. The *setroute* parameter allows automatically setting the default route for the outside interface. The MTU on the outside interface is automatically set to 1492, which is the correct setting to provide PPPoE encapsulation. It is also possible to designate a fixed IP address for the outside interface. The PIX still has to provide the ISP with the correct username and password to establish the session:

```
PIX1 (config) # ip address outside 1.2.3.4 255.255.255.0 pppoe
```

It is possible to use the *dhcp auto\_config* command if you run the DHCP server on PIX to pick up DNS and WINS settings from your provider via the PPPoE client:

```
PIX1 (config) # dhcpd auto_config outside
```

To monitor and troubleshoot the PPPoE client, use the following commands:

```
show ip address outside pppoe
debug pppoe event | error | packet
show vpdn session pppoe [id <sess_id>|packets|state|window]
```

Examples of output are as follows:

```
PIX1 (config) # show vpdn
Tunnel id 0, 1 active sessions
time since change 10240 secs
Remote Internet Address 10.0.1.1
Local Internet Address 192.168.2.254
1006 packets sent, 1236 received, 98761 bytes sent, 123765 received
Remote Internet Address is 10.0.1.1
Session state is SESSION_UP
Time since event change 10237 secs, interface outside
PPP interface id is 1
1006 packets sent, 1236 received, 98761 bytes sent, 123765 received
PIX1 (config) # show vpdn tunnel
PPPoE Tunnel Information (Total tunnels=1 sessions=1)
Tunnel id 0, 1 active sessions
time since change 10240 secs
Remote Internet Address 10.0.1.1
```

```

Local Internet Address 192.168.2.254
1006 packets sent, 1236 received, 98761 bytes sent, 123765 received
PIX1(config)# show vpdn session
PPPoE Session Information (Total tunnels=1 sessions=1)
Remote Internet Address is 10.0.1.1
Session state is SESSION_UP
Time since event change 100238 secs, interface outside
PPP interface id is 1
1006 packets sent, 1236 received, 98761 bytes sent, 123765 received

```

## EasyVPN

The purpose of EasyVPN is to simplify the management of VPN deployments. This is accomplished by centralizing the point of configuration for a VPN environment to the EasyVPN server, which then pushes the policies out to EasyVPN clients. EasyVPN is supported across a variety of Cisco platforms, including the PIX firewall, which supports both the EasyVPN client and server.

### NOTE

---

Since EasyVPN client is only supported on the PIX501 and 506E, which are not currently compatible with version 7.0, it is only possible to configure EasyVPN server in version 7.0.

---

## EasyVPN Server

To configure the PIX firewall as an EasyVPN server, first complete all the usual configuration necessary for the PIX to establish IP communication with its inside and outside networks. Next, add the standard VPN configuration elements, including IPsec transform sets and dynamic crypto maps, as well as applying the crypto map to the outside interface. Be sure to configure the phase 1 ISAKMP parameters as you would for any other VPN connection.

The configuration parameters specific to an EasyVPN server are specified with the *vpn-group* command. All configuration defined will automatically be pushed out and accepted by an EasyVPN client that connects to the PIX as an EasyVPN server. Here are some commonly used EasyVPN parameters:

```

PIX1 (config) # vpngroup testgroup1 address-pool mypool1
PIX1 (config) # vpngroup testgroup1 idle-time 1200
PIX1 (config) # vpngroup testgroup1 password mypass1
PIX1 (config) # vpngroup testgroup1 dns-server 192.168.1.1

```

```
PIX1 (config) # vpngroup testgroup1 wins-server 192.168.2.1
PIX1 (config) # vpngroup testgroup1 default-domain mydomain.com
```

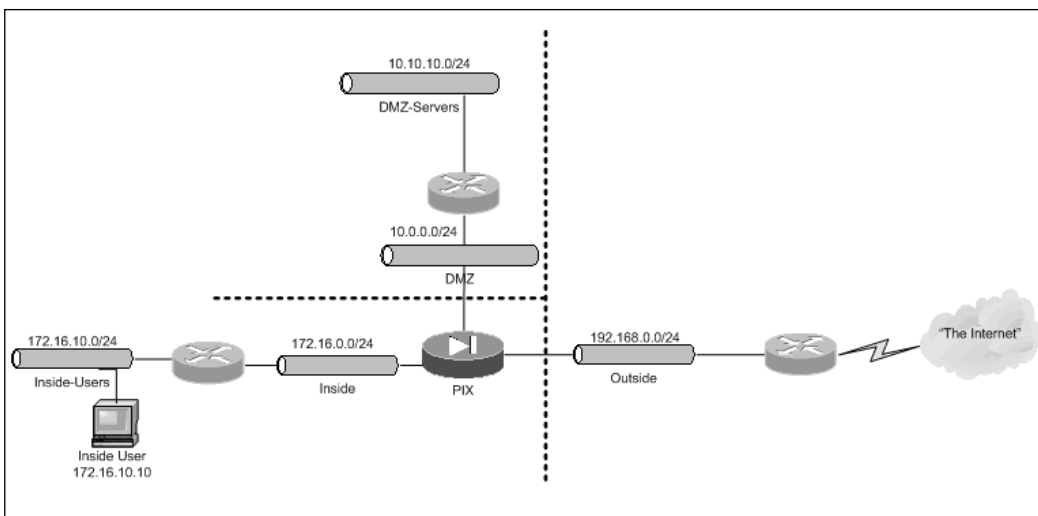
Here, we are configuring the EasyVPN server to use the previously defined address pool mypool1, with an idle timeout of 1200 seconds, a password of mypass1, DNS and WINS servers of 192.168.1.1 and 192.168.2.1, respectively, and a default domain of mydomain.com. As you can see from this example, configuring such parameters centrally, rather than on each individual remote client, has great advantages in terms of efficiency. This is assuming that these parameters will be consistent across your network, including VPN remotely connected devices, but this is often the case.

## Routing and the PIX Firewall

The PIX firewall can do a limited amount of routing. Let us not forget that it is a firewall, not a router. Its mission is to secure the network, not find the best path to a destination. Having said that, Cisco has incorporated a limited set of routing features into the PIX. As of this writing, the PIX can route via static routes, RIP, and OSPF, and, oddly enough, via network address translation.

In this discussion of routing, we will use the network architecture in Figure 7.2 to guide our efforts. Our goal with this architecture is to enable all three of the networks (and points behind them) to reach other.

**Figure 7.2** Routing with the the PIX



## Unicast Routing

Unicast routing, as opposed to multicast routing, which we will cover later, includes static and dynamic routing, both of which are supported by the PIX. Static routing is most appropriate for cases where the PIX has only one or unchanging routes to its destinations. In

these cases, it does not make sense to add the complexity and overhead of a dynamic routing protocol. In contrast, if there are multiple possible routes to a destination, such as two or more redundant paths to the same destination, and the PIX must evaluate which path to take, a dynamic routing protocol may be more appropriate.

## Static Routes

The use of static routes has been a mainstay of PIX routing since its inception. It is the oldest and most static method of routing of the PIX firewall—make of that what you will. This command is largely unchanged from the previous versions. While similar to its IOS counterpart (specify the network and its next hop/interface), it is still obviously a PIX command.

The command is simple, with few options, as shown:

```
PIX1(config)# route ?
```

```
Current available interface(s):
```

```
DMZ      Name of interface Ethernet2
Outside  Name of interface Ethernet0
inside   Name of interface Ethernet1
```

```
PIX1(config)# route DMZ ?
```

```
Hostname or A.B.C.D  The foreign network for this route, 0 means default
```

```
PIX1(config)# route DMZ 0.0.0.0 ?
```

```
A.B.C.D  The netmask for the destined foreign network
```

```
PIX1(config)# route DMZ 0.0.0.0 1.1.1.1 ?
```

```
Hostname or A.B.C.D  The address of the gateway by which the foreign
network is reached.
```

```
PIX1(config)# route DMZ 0.0.0.0 1.1.1.1 1.1.1.1 ?
```

```
<1-255>  Distance metric for this route, default is 1
```

```
tunneled  Enable the default tunnel gateway option, metric is set to 255
```

For our architecture in Figure 4.2, we need to enter the following commands to have full routing. Notice what interface each static route references.

```
route Outside 0.0.0.0 0.0.0.0 192.168.0.10 10
route inside 172.16.10.0 255.255.255.0 172.16.0.10 1
route DMZ 10.10.10.0 255.255.255.0 10.0.0.10 1
```

The first route in this example is a default route. A default route is simply a static route that includes all possible IP addresses, or 0.0.0.0 as the IP address and 0.0.0.0 as the netmask.

## RIP

RIP takes the PIX firewall a few steps in the direction of automating routing on the PIX firewall. RIP is a distance vector routing protocol that suffers from a small routing diameter (anything over 15 hops is unreachable). Its use on the PIX firewall is mainly limited to immediate networks and immediate neighbors. Chances are, any neighbors of the PIX will have to run an additional routing protocol such as EIGRP or OSPF in order to expand the routing diameter.

The PIX supports both RIP versions 1 and 2; the latter fixes many problems of the former by adding support for variable length subnet masks (VLSM) and authentication for security.

Harkening back to our architecture in Figure 4.2., the following RIP configuration would enable routing between the networks.

```
PIX1(config)# rip DMZ default ?
    version  RIP version, default is RIPv1
    <cr>

PIX1(config)# rip DMZ default version 2 ?
    authentication  Authenticate using the specified mode
    <cr>

PIX1(config)# rip DMZ default version 2 authentication ?
    md5      Authenticate using md5 mode
    text     Authenticate using text mode

PIX1(config)# rip DMZ default version 2 authentication md5 ?
    WORD < 17 char  The shared key to be used for authentication

PIX1(config)# rip DMZ default version 2 authentication md5 syngress ?
    <0-255>  The shared key id that matches the key

PIX1(config)# rip DMZ passive ?
    version  RIP version, default is RIPv1
    <cr>

PIX1(config)# rip DMZ passive version ?
    1  RIP Version 1 (RIPv1)
    2  RIP Version 2 (RIPv2)

PIX1(config)# rip DMZ passive version 2 ?
    authentication  Authenticate using the specified mode
```

```

<cr>

PIX1(config)# rip DMZ passive version 2 authentication ?
  md5    Authenticate using md5 mode
  text   Authenticate using text mode

PIX1(config)# rip DMZ passive version 2 authentication md5 ?
  WORD < 17 char  The shared key to be used for authentication

PIX1(config)# rip DMZ passive version 2 authentication md5 syngress ?
  <0-255>  The shared key id that matches the key

PIX1(config)# rip DMZ passive version 2 authentication md5 syngress 255

rip outside passive version 2 authentication md5 thisisakey 2
rip outside default version 2 authentication md5 thisisakey 2
rip inside passive version 1
rip dmz passive version 2

```

We want our PIX firewall to transmit a route to its peers out of each of its interfaces, with the exception of the Outside interface where we just want it to learn RIP routers and populate its tables. Therefore, our configuration will be:

```

rip DMZ default version 2 authentication md5 syngress 255
rip inside default version 2 authentication md5 syngress 255
rip DMZ passive version 2 authentication md5 syngress 255

```

Notice that we are running version 2, as are our router neighbors, and that our router neighbors are injecting default routes, as well as routes from beyond. This will enable the PIX to reach these destinations.

## OSPF

Support for the Open Shortest Path First (OSPF) on the PIX firewall has been around since version 6.3. OSPF is a link state routing protocol that uses costs to determine the best (shortest) path to a network. OSPF operates by each router placing itself at the center of the network and hoarding and updating information about the links in the topology of the autonomous systems.

OSPF more than overcomes the limitations of RIP. It is a very robust, very fast routing protocol. It does require more planning and more attention to design, as it operates in a stricter hierarchy.

To accomplish our routing requirements for the network in Figure 7.2, we need to deploy the following configuration on our PIX firewall. If we assume that we are dealing with a single area at this time, we could have multiple areas.

```
router ospf 100
 network 10.0.0.0 255.255.255.0 area 0
 network 172.16.0.0 255.255.255.0 area 0
 network 192.168.0.0 255.255.255.0 area 1
 log-adj-changes
 default-information originate always
```

## Network Address Translation as a Routing Mechanism

With a creative design, you can use NAT to handle some of your routing needs. For the network in Figure 7.2, we have deployed NAT on each neighbor router, translating all source addresses heading to the PIX firewall to something on the subnet on each interface.

The PIX operates in blissful ignorance; it does not know about the networks behind each router. As far as it is concerned, there are only three networks. Because they are all connected, the PIX firewall can simply direct traffic to the appropriate interface, without using a static route, or dynamic routing protocol. The configuration required for this to work involves configuring NAT on the router attached to each PIX interface, which is beyond the scope of PIX configuration, and therefore this book. However, the key factor to keep in mind is that because the remote routers are translating all traffic to the IP addresses that are directly connected to the PIX, the PIX will therefore have connected routes to these networks by default. It is for this reason that the PIX does not require any static or dynamic routing in this case.

The only caveat or requirement for routing in this scenario is that you might want to or need to add a default route to handle traffic such as that that going to the Internet.

## Multicast Routing

While unicast packets are sent from one source to one destination, multicast packets are sent from one source to multiple destinations. This is useful for applications that are designed to distribute information to two or more users, and especially to higher volumes of users, where sending individual copies of the data from the source to each destination can quickly create a congested network. Multicast, on the other hand, need not duplicate the same data along any one network segment; if two or more hosts on the same network segment are to receive a multicast stream of data, only one copy of the data needs to be sent by the source host.

A good example of an application that benefits from multicast is Microsoft Netmeeting. With Netmeeting, typically one presenter is transmitting information to multiple remote users. Rather than sending the data of the presentation individually to each user, multicast allows for a significant improvement in network efficiency.

Multicast traffic makes use of a defined set of IP addresses: 224.0.0.0 through 239.255.255.255. Many of these are reserved on the Internet (see [www.iana.org/assignments/multicast-addresses](http://www.iana.org/assignments/multicast-addresses)).

Multicast routing is supported by the PIX firewall, in order to allow it to participate in a multicast network where routing is required for multicast traffic to reach its destination. The PIX supports stub multicast routing and PIM multicast routing. No matter which type of multicast routing you want to enable, the first step to enabling multicast routing support on the PIX is to enter this command:

```
PIX1(config)# multicast-routing
```

Once multicast routing is enabled globally, the PIX is ready to build a multicast routing table. This table's size is only restricted by the amount of memory installed in your PIX firewall. Table 7.1 describes the maximum number of routes the PIX can store based on its memory. This table is courtesy of Cisco (see [www.cisco.com/en/US/partner/products/sw/secursw/ps2120/products\\_configuration\\_guide\\_chapter09186a00804231c6.html#wp1041648](http://www.cisco.com/en/US/partner/products/sw/secursw/ps2120/products_configuration_guide_chapter09186a00804231c6.html#wp1041648)).

**Table 7.1** PIX Multicast Routing Table Size Limits

Table	16MB	128MB	128+MB
MFIB	1000	3000	5000
IGMP Groups	1000	3000	5000
PIM Routes	3000	7000	12000

Even though you have enabled multicast routing on the PIX, it is still possible to specify whether each interface should participate in multicast routing. This is done by enabling or disabling IGMP on a per-interface basis. Since IGMP is the protocol used by hosts to report group membership, disabling IGMP on an interface effectively disables multicast routing functionality on that interface. Note that by default, enabling multicast routing on the PIX enables IGMP on all interfaces. To disable IGMP on a specific interface, for example ethernet0, enter the following commands:

```
PIX1(config)# int ethernet0
PIX1(config-if)# no igmp
```

To re-enable IGMP on an interface where it was previously disabled, enter the following:

```
PIX1(config)# int ethernet0
PIX1(config-if)# igmp
```

## Stub Multicast Routing

Stub multicast routing is a method of multicast routing that simply involves forwarding IGMP messages to another multicast-enabled router, rather than fully participating in the multicast routing protocol. This is an appropriate method to use in such cases where you want to centralize your multicast routing to your uplink router, and do not need to push full multicast functionality to other nodes, such as the PIX, but still want hosts behind the PIX to be able to function in a multicast environment.

Stub multicast routing is accomplished by configuring the PIX to act as a route proxy. To configure the PIX firewall to forward IGMP messages from one interface to another, for example from ethernet1 to ethernet0 (outside), enter the following commands:

```
PIX1(config-if)# int ethernet1
PIX1(config-if)# igmp forward interface outside
```

## PIM Multicast Routing

An alternative to stub multicast routing is PIM (note that they cannot both be enabled concurrently). PIM stands for Protocol Independent Multicast, and is not dependent on a specific unicast routing protocol—it can function as long as IP connectivity is established through whatever static or dynamic unicast protocol is in use. By enabling PIM on the PIX, it is able to fully participate in multicast routing, which is appropriate in cases where there are hosts behind the PIX that make use of multicast applications, and there are multiple upstream multicast destinations for the PIX to route traffic to.

By default, when multicast routing is enabled on the PIX, PIM is also enabled on all interfaces. To disable PIM on an interface, for example ethernet0, enter the following commands:

```
PIX1(config-if)# int ethernet0
PIX1(config-if)# no pim
```

To re-enable PIM on an interface where it was previously disabled, enter the following:

```
PIX1(config-if)# int ethernet0
PIX1(config-if)# pim
```

Although there are three rendezvous-point (RP) modes—static RP, auto-RP, and BSR—the PIX firewall only supports static RP. To configure the PIX for a static RP, enter the following command:

```
PIX1(config)# pim rp-address 10.1.1.1
```

## BGP through PIX Firewall

Allowing BGP through the PIX firewall is quite straightforward. Assuming you have configured the PIX to enable IP connectivity between the BGP routers, the only additional step is to allow the BGP traffic via an access-list. For example, to allow BGP traffic from a router with IP address 172.16.0.1 to another router with IP address 172.16.1.1, enter the following command:

```
PIX1(config)# access-list bgp1 permit tcp host 172.16.0.1 host 172.16.1.1 eq bgp
```

You will then need to apply this access-list to the appropriate interface. For example, to apply it to the outside interface, enter this command:

```
PIX1(config)# access-group bgp1 in interface outside
```

# Queuing and Policing

Queuing and policing are part of the Quality of Service (QoS) functionality built in to the PIX. QoS, simply explained, is the prioritization of some traffic over other traffic, for the sake of providing the best service to the traffic that is most important or sensitive to network degradation.

The first step to enabling QoS on the PIX is to identify the traffic you would like to prioritize. Identifying traffic is done with class maps. Class maps can match traffic based on an access-list, a DSCP value, a TCP or UDP port, an IP destination, an IP precedence, an RTP port, or a tunnel group. Here is an example of a class map that matches all HTTP packets:

```
PIX1(config)# access-list allhttp permit tcp any any eq 80
PIX1(config)# class-map http-map
PIX1(config-cmap)# match access-list allhttp
```

Once the traffic you are interested in is identified by a class map, the next step is to use a policy map to apply an action to this traffic. Rate limiting, also known as policing, is applied this way. For example, to limit HTTP traffic to 100,000 bits per second, with a 10,000 bytes per second burst size, enter the following commands:

```
PIX1(config)# policy-map httplimit
PIX1(config-pmap)# class http-map
PIX1(config-pmap-c)# policy outside 100000 10000
```

The final step is to enable the policy, either globally or on a specific interface. For example, to enable the previously defined HTTP policy on the outside interface of the PIX, enter the following command:

```
PIX1(config)# service-policy httplimit interface outside
```

To enable this policy on all interfaces rather than the outside interface:

```
PIX1(config)# service-policy httplimit global
```

## Summary

The PIX supports a number of services that are designed to augment the value of the PIX beyond its core firewall functionality. DHCP allows the PIX to provide IP address and related assignment without the need for a separate server. The PIX's QoS functionality may be sufficient for your traffic prioritization needs, eliminating the need for another router that would otherwise be required. Furthermore, the routing functionality that is built in to the PIX provides a flexible set of unicast and multicast routing options that allow the PIX to establish full IP connectivity to the rest of the network.

Finally, with the PIX's EasyVPN functionality, it is capable of acting as an EasyVPN server, where it can be configured to propagate a variety of policy settings to any number of EasyVPN clients.

## Solutions Fast Track

### DHCP Functionality

- ☑ The Cisco PIX firewall can act as both a DHCP server and a client. PIX DHCP features are best suited for small networks because they have some limitations—for example, a DHCP server can support a maximum of 256 clients. There is also no BOOTP support and no failover support.
- ☑ The DHCP client can be configured only on the outside interface. It is able to obtain an IP address, subnet mask, default route, and DNS and WINS settings from the server. The obtained address can be used for NAT or PAT on the outside interface.
- ☑ The DHCP server can be configured only on the inside interface and serves only directly connected clients. The number of active clients is dependent on the PIX model and software version. It is possible to pass some settings that are obtained by PIX DHCP clients from the outside interface to the DHCP server running on the inside interface.

### PPPoE

- ☑ PPPoE allows the PIX to sit on the remote end of an Internet connection that makes use of this protocol to dynamically assign an IP address and other parameters.
- ☑ The PIX supports multiple PPPoE authentication types, including PAP and CHAP.

## EasyVPN

- ☑ EasyVPN allows for a central point of configuration for multiple remote VPN endpoints, for a variety of policy parameters.
- ☑ The PIX501 and 506E support the EasyVPN client, but these platforms are not supported by version 7.0.
- ☑ All other PIX firewalls may act as an EasyVPN server.

## Routing and the PIX Firewall

- ☑ The PIX supports static and default routes.
- ☑ For more complex routing situations, the PIX can be configured for dynamic unicast routing with either RIP or OSPF.
- ☑ The PIX also supports multicast routing, with either stub multicast routing or PIM.

## Queuing and Policing

- ☑ Traffic must first be classified with a class map. Traffic may be identified by a number of parameters, including access-lists or DSCP values.
- ☑ Action can be taken on the identified traffic with a policy map. Policing can be applied in this case to limit the amount of bandwidth allocated.
- ☑ The policy map is applied to a single interface or all interfaces with the *service-policy* command.

## Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to [www.syngress.com/solutions](http://www.syngress.com/solutions) and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at [ITFAQnet.com](http://ITFAQnet.com).

**Q:** What is the advantage to running my DHCP server on my PIX firewall rather than on a stand-alone server?

**A:** The main advantage is simplifying your network infrastructure by having fewer hosts. Combining firewall and DHCP services onto the same PIX firewall allows you to manage only that one device rather than having to worry about managing and maintaining a separate server.

**Q:** How do I know if I should use static or dynamic routing?

**A:** In general, you should always choose the simplest routing solution that satisfies your needs. Therefore, static routing is preferred over dynamic routing, since it has lower overhead and less complexity. In cases where you have a network where there are a small and unchanging number of routes to each destination, static routing should be sufficient. In other cases, when the PIX needs to make a routing decision based on other factors, you will need to use one of its supported dynamic routing protocols.

**Q:** When should I enable queuing and policing?

**A:** Quality of Service, including queuing and policing, are generally needed when you have multiple types of traffic on your network, and some types of traffic need to have priority over others. A good example of this is a network with voice or video traffic; since voice and video are very sensitive to delay, they need to be given priority over other types of traffic. In this case, you should enable queuing to prioritize the voice and video traffic.

**Q:** When should I use EasyVPN?

**A:** EasyVPN is most useful when you have many endpoints, and you wish to push the policy settings out to them rather than configuring them all manually. This is most often appropriate when you control all the endpoints. If a third-party controls an endpoint, they may wish to manually control the policy settings of that device rather than have you push out configuration from your EasyVPN server.