Defense Plan 3: The Internal “Pseudo” DMZ

Defensive Tactics in this Chapter:

- Using TCP/IP Filtering (TCP/IP Security)
- Using Routing and Remote Access Service (RRAS) Packet Filters
- Using IPSec Policies

☑ Summary
☑ Defensive Tactics Fast Track
☑ Frequently Asked Questions
Chapter 4 • Defense Plan 3: The Internal “Pseudo” DMZ

Introduction

In the last two chapters, we went over how you create DMZ segments using untrusted networks. In the trihomed DMZ configuration, the untrusted segment is a public address segment directly connected to the ISA server and is a subnet of your public address block. In the back-to-back DMZ configuration, you had the choice of using public or private addresses. In all cases, the DMZ segment was an untrusted network segment. Untrusted network segments are not in the LAT.

Many organizations have only a small number of public IP addresses at their disposal. Creating a public address DMZ is out of the question for these organizations because they need to use their few public IP addresses to support Web and server publishing rules. Even if you have a large number of IP addresses at your disposal, you might not want to get another Windows 2000 or Windows .Net Server and a second copy of ISA Server to create either the public or private address back-to-back DMZ configuration.

There is a way for you to create what could be considered a “pseudo-DMZ” out of a trusted network segment. The trusted internal network segment is contained in the LAT. You can install multiple internal interfaces on the ISA Server computer. Unlike the single external interface limitation that ISA Server has, there is no limit to internal or DMZ interfaces.

You can place two or more network adapters on the ISA server that are dedicated to LAT network segments (Figure 4.1). Normally, ISA Server will route packets between these two network segments when IP routing is enabled on the ISA server via the IP Packet Filters Properties dialog box (Figure 4.2). The problem with this configuration is that ISA Server will route all packets between LAT segments. ISA Server sees all LAT segments as trusted network segments and therefore does expose these packets to ISA Server access policies.

This is acceptable when all LAT segments are trusted, but it doesn’t work if you want to create a DMZ out of a LAT segment. Why would you want to do so? The primary reason is that you don’t have enough public IP addresses to create a trihomed DMZ. Another reason would be that you want to take advantage of Web and server publishing rules, like you can with a private address back-to-back DMZ, but you can’t afford to implement a second ISA server. If you find yourself in this situation, you need to create a LAT-based DMZ.

It’s vital that you control what traffics moves between the LAT-based DMZ (LATDMZ) and other LAT networks. If you can accomplish that goal, you can have the same security with a LATDMZ as you would with any other type of DMZ segment.
There are three techniques that you can use to create the LATDMZ segment:

- TCP/IP Security (TCP/IP filtering)
- Routing and Remote Access (RRAS) packet filters
- IPSec policies
TCP/IP Security is the simplest method but has limited utility; RRAS packet filters are less simple but more powerful and flexible than TCP/IP Security filters; and IPSec policies can become quite complicated but provide a great deal of flexibility and customization. With increasing complexity comes greater flexibility. You’ll do yourself a favor by learning about each technique and then implementing the one(s) that best suit(s) your needs.

**ISA Server Alert**

As we go through the techniques you use to create and support a LATDMZ segment, it’s important to consider the primary purpose of a DMZ segment, which is to segregate your security zones. The practical application of this principle means that you do not want any hosts on your DMZ segment to have any dependencies on security principles on the internal network (for example, the Active Directory). DMZ hosts must be considered bastion hosts, and as bastion hosts, they have a relatively high probability of being compromised. If you are compelled to extend your internal network’s security zone into the DMZ, move the server that is dependent on the internal network’s directory services out of the DMZ and into the internal network. Configuration will be more straightforward, and you lessen the risk of other hosts on the DMZ becoming potential launch points for an internal network attack.

**Using TCP/IP Filtering (TCP/IP Security)**

TCP/IP filtering is a simple filtering method that you can use to control inbound access to servers on the LATDMZ segment. You can use TCP/IP filtering to control which TCP, UDP, or IP protocols have inbound access to the servers on the DMZ segment. TCP/IP filtering does not provide a way for you to control outbound access from the servers on the LATDMZ, nor does it provide a way for you to allow some clients inbound access to the LATDMZ host. TCP/IP filtering does not allow you to control outbound access from a LATDMZ host to an internal network client.

It might seem that TCP/IP filters wouldn’t be of much use on a LATDMZ host. After all, you’re using Web or server publishing rules to control inbound access to the hosts on the LATDMZ, so there should be no reason for this type of additional security. You would expect that the only Internet traffic that can reach the LATDMZ hosts is traffic that you explicitly allow using Web and server publishing rules.

While Web and server publishing rules allow you to control the traffic coming from the Internet, it’s still possible that one of the servers on the DMZ could be compromised either by an Internet-based host or someone who was able to connect to the
LATDMZ from the internal network or from another location (such as a network management segment or other “out of band” connection). TCP/IP filters give you added protection.

Another advantage of TCP/IP filtering is that it controls inbound access to LATDMZ hosts using a kernel-mode process. Unlike the RRAS packet filters and IPSec policies that we’ll cover later in this chapter, TCP/IP filtering does not depend on user-mode processes or the Workstation/Server services. Security on user-mode services can be interrupted by various exploits such as buffer overruns that leave the server running and thus accessible to the intruder; cracking kernel-mode security processes is more complicated, and typically would end up “blue screening” the server. While this causes a denial of service (DoS) situation on the compromised host, it doesn’t allow the intruder to use the compromised server as a fulcrum for a subsequent attack against the internal network.

TCP/IP filtering on a LATDMZ host provides a great layered security configuration when coupled with either or both RRAS packet filters and IPSec policies. However, there are some limitations to TCP/IP filtering of which you should be aware:

- You cannot filter ICMP traffic with TCP/IP filtering.
- You cannot filter outbound traffic using TCP/IP filtering.
- You cannot configure selective access to particular computers/subnets using TCP/IP filtering; the filter applies to all inbound traffic equally.
- You cannot block all UDP or TCP protocols by blocking IP protocols 6 and 17; you can block all UDP and/or TCP protocols by using an exception list.

With these limitations in mind, let’s look at how to configure TCP/IP Filtering.

**Configuring TCP/IP Filtering**

Perform the following steps to configure TCP/IP filtering:

1. Open the **Control Panel** and then open the **Network and Dial-up Connections** applet (Figure 4.3).
2. Right-click on the interface you need inbound access control and click **Properties**.
3. Select the **Internet Protocol (TCP/IP)** and click **Properties** (Figure 4.4).
4. Click **Advanced** in the **Internet Protocol (TCP/IP) Properties** dialog box (Figure 4.5).
5. Click the **Options** tab in the **Advanced TCP/IP Settings** dialog box.
Figure 4.3 The Network and Dial-Up Connections Window

Figure 4.4 The Local Area Connections Dialog Box

Figure 4.5 The Internet Protocol (TCP/IP) Properties Dialog Box
6. Click the **TCP/IP filtering** entry and click **Properties** (Figure 4.6).

**Figure 4.6** The Advanced TCP/IP Settings Dialog Box

7. Place a check mark in the **Enable TCP/IP Filtering (All adapters)** check box. Note that doing so enables filtering for *all* adapters, but you configure the filters on a per-adapter basis. The same filters do *not* apply to all adapters (Figure 4.7).

**Figure 4.7** The TCP/IP Filtering Dialog Box

8. There are three columns: **TCP Ports**, **UDP Ports**, and **IP Protocols**. For each, you have two choices: **Permit All** and **Permit Only**. If you want to permit all packets for TCP or UDP traffic, then leave the **Permit All** option selected. If you want to allow selected TCP or UDP traffic, select the **Permit Only** option, click **Add**, and enter the appropriate port in the **Add Filter**
dialog box. In Figure 4.8, we allow only TCP ports 25 and 80 (SMTP and HTTP) and GRE (Generic Routing Encapsulation for PPTP–IP Protocol 47) inbound. All other inbound packets (except ICMP) will be dropped.

**Figure 4.8 Configuring Ports in the TCP/IP Filtering Dialog Box**

9. If you want to block all UDP or all TCP traffic, select the **Permit Only** option and do not enter any port numbers in the **UDP Ports** or **TCP Port** column. You cannot block all UDP or all TCP traffic by selecting the **IP Protocols Permit Only** option and exclude IP Protocol 6 (TCP) and IP Protocol 17 (UDP). In Figure 4.9, we allow only inbound GRE. All other inbound packets are dropped except for ICMP. Note that if you wanted to allow inbound PPTP connection, this filter wouldn’t be enough; you would have to add a filter to allow TCP 1723.

**Figure 4.9 Permitting Only GRE Packets Inbound**

10. You cannot block ICMP messages. This is true even if you select the **IP Protocols** column and select **Permit Only** and fail to include IP Protocol 1.
Internet Protocol Number assignments are listed in Microsoft Knowledge Base article Q289892. You can find a long list of TCP and UDP port number assignments at www.iana.org/assignments/port-numbers.

Summary of TCP/IP Filtering

TCP/IP filtering is a good first step in protecting your servers in the DMZ from external network hosts, other hosts on the LATDMZ, and hosts on the internal network. As useful as TCP/IP filtering might be, it’s only a piece of your LATDMZ defense plan. You need to add RRAS packet filters and/or IPSec policies to make your LATDMZ a true DMZ segment.

Using Routing and Remote Access Service (RRAS) Packet Filters

The Routing and Remote Access Service (RRAS) is part of every Windows 2000 and Windows .NET Server installation. RRAS is part of the default installation and there is no method to exclude it during setup. Although the service is installed, it is not enabled.

You don’t have to run RRAS on the ISA Server machine. ISA Server can protect your DMZs and LAT segments without RRAS enabled. You can even route packets between LAT segments directly connected to the ISA server without RRAS enabled; just enable IP Routing in the IP Packet Filters Properties dialog box (Figure 4.10).

Figure 4.10 The IP Packet Filters Properties Dialog Box
The problem with the built-in IP routing between LAT segments directly connected to the ISA server is that you can’t control traffic moving between the segments. Although ISA Server controls traffic moving between the external interface and the LAT segments, it trusts all traffic moving between the LAT segments and lets it pass uninspected.

That’s fine for non-DMZ situations, but in our LATDMZ scenario, we need total control over what is sent from the LATDMZ segment to the internal network. We would also like to control traffic moving from the internal network to the DMZ segment.

The good news is that you can exercise a great deal of control over what packets move between the LATDMZ segment and the internal network using RRAS packet filters. We can use ISA Server site and content and protocol rules to control outbound access to the Internet from DMZ hosts, but we need to use another method to control what happens between the LATDMZ and internal network segments. RRAS packet filtering is one of those methods.

You have to start RRAS before you can use RRAS packet filters. You’ll likely find yourself in one of the following three scenarios:

- RRAS was enabled before you installed ISA Server.
- ISA Server is installed, and you haven’t started RRAS yet.
- You enabled RRAS when you ran one of the ISA Server VPN wizards.

If RRAS was enabled before you installed ISA Server, you’ve already had to deal with the service dependency problems that crop up when you don’t let the ISA server enable RRAS. Hopefully you’ve resolved the problem, but if not, here’s the fix:

1. You need to make the Routing and Remote Access Service dependent on the ISA Server Control Service (\isactrl). To do this, you must edit the Registry.
2. Open the Run command from the Start menu. In the Open text box, type \regedt32 and click OK.
3. Navigate to the following key:
   \HKLM\System\CurrentControlSet\Services\RemoteAccess
4. Double-click on the DependOnService value and add the isactrl string in the Multi-String Editor (Figure 4.11).

Run this fix whenever you start RRAS independently of the ISA Server VPN wizards. You might find that the ISA Server services won’t start up properly or automatically if you don’t let ISA Server start RRAS.
You're in good shape if you haven’t started RRAS and ISA Server is installed. We recommend that you let the ISA Server VPN wizards start RRAS. You don’t have to allow VPN connections. If you want to block incoming VPN connections, all you need to do is disable the VPN protocol packet filters that the VPN wizard creates to allow VPN connections into the network.

Perform the following steps to let the ISA server enable RRAS for you:

1. In the ISA Management Console, expand your server name and then right-click on the Network Configuration node. Click on the Allow VPN Connections command.
2. Click Next on the Welcome to the ISA Virtual Private Network Configuration Wizard page.
3. Click Finish on the Completing the ISA VPN Server Configuration Wizard page.
4. A dialog box will appear and ask if you want to start RRAS. Click Yes.

After RRAS is started, go to the ISA Server Management console and you’ll see the following packet filters in the IP Packet Filters node:

- Allow L2TP protocol IKE packets
- Allow L2TP protocol packets
- Allow PPTP protocol packets (client)
- Allow PPTP protocol packets (server)

If you don’t want the ISA server to act as a VPN server, right-click on each of these packet filters and click Disable.

Configuring RRAS Packet Filters

RRAS packet filters allow you to control traffic arriving and leaving a particular interface on the RRAS computer. In the case of the LAT-based DMZ, we are concerned
with controlling packets moving between the LATDMZ segment and the internal network. The ISA Server software will control what moves between the LAT segments and the Internet.

You should rename the network interfaces before working with RRAS packet filters. Doing so will make it easier to identify which interface is the LATDMZ interface, the internal network interface, and the external network interface.

Perform the following steps to change the names of the adapters:

1. Open the Network and Dial Up Connections window. This window contains all the network interfaces installed on your computer.
2. Right-click on the LATDMZ interface and click the Rename command. Type in the name LATDMZ and press Enter.
3. Right-click on the internal interface and click the Rename command. Type in the name LAN and press Enter. Do not name the internal interface INTERNAL. That name is already assigned to the virtual interface that the RRAS server uses for VPN connections.
4. Right-click the external interface and click the Rename command. Type in the name EXTERNAL. You should see what appears in Figure 4.12.

**Figure 4.12 Renaming the ISA Server Interfaces**

5. Now open the Routing and Remote Access console from the Administrative Tools menu. Expand your server name and then expand the IP Routing node in the left pane of the Routing and Remote Access console. Click on the General node and you should see something like what appears in Figure 4.13.
6. In the right pane you’ll see the names you assigned to the interfaces. Double-click on your LATDMZ interface and you’ll see what appears in Figure 4.14.

**Figure 4.14 The LATDMZ Interface Properties Dialog Box**
How RRAS Packet Filters Work with a LATDMZ Segment

RRAS packet filters work on an "exception" basis. You can allow all packets except for those for which you create a filter, or you can deny packets except for those for which you create an exception. This exception-based access control makes inbound and outbound access control to and from the LATDMZ segment challenging.

For example, suppose you want to prevent any packets from moving from a host on the DMZ segment to any host on the internal network. The goal is to prevent traffic sourcing on the LATDMZ from reaching the internal network.

In this example, our LATDMZ segment is on network ID 172.16.0.0/24. The internal network is on network ID 10.0.0.0/24. Follow these steps to prevent all traffic originating on the DMZ segment from entering the internal network:

1. In the Routing and Remote Access console, expand the server name, and then expand the IP Routing node in the left pane of the console. Click on the General node and then double-click on the LATDMZ interface in the right pane of the console.

2. In the LATDMZ Properties dialog box, click Input Filters.

3. In the Input Filters dialog box, click Add.

4. In the Edit IP Filter dialog box, put check marks in both the Source network and Destination network check boxes. In the IP address text box for the Source network, put in the network ID for the LATDMZ segment in the IP address text box, and put the subnet mask for the LATDMZ segment in the Subnet mask text box (Figure 4.15). In the IP address text box for the Destination network, put in the network ID that summarizes the internal network, and put in the subnet mask for that network ID in the Subnet mask text box. In the Protocol drop-down list box, select Any. Doing so applies the filter to all protocols. Click OK.

5. In the Input Filters dialog box (Figure 4.16), make sure that the Receive all packets except those that meet the criteria below option is selected. The input filter applies to all packets arriving to the interface. This filter allows all packets to be accepted by the LATDMZ interface except those that match the filter you created. This filter allows all packets to move between the LATDMZ segment and the Internet, but no packets are accepted from hosts on the DMZ segment if the destination network ID in the packet is that of the internal network. This prevents any host on the DMZ segment from accessing any resource on the internal network. Click OK.
6. Click **Apply** and then click **OK** in the **LATDMZ Properties** dialog box.

This packet filter completely isolates the DMZ segment from the internal network, while still allowing the ISA server to move packets between the DMZ segment and the Internet. You’ll be able to create Web and server publishing rules to make servers on the LATDMZ available to Internet hosts. The only difference between the LATDMZ and the internal network is that no direct communications take place between the LATDMZ and the internal network.

You’re very limited in terms of the granularity of access control obtainable using RRAS packet filters. The ideal setup would allow you to prevent outgoing connections from the LATDMZ to the internal network except for a handful of protocols you want
to allow. For example, you might want to allow an SMTP relay server on the LATDMZ to send SMTP messages to an SMTP server on the internal network.

Why can’t you allow this type of selective traffic? Think about the packet filters needed to allow TCP port 25 traffic from the DMZ segment into the internal network while at the same time blocking all other traffic.

You could create a packet filter for TCP port 25 with the source network ID being the DMZ segment’s network ID, and the destination network ID the internal network’s network ID, and use the Drop all packets except those that meet the criteria below option. However, this would prevent packets from moving between DMZ segment hosts and the Internet! You wouldn’t be able to publish any servers on the DMZ segment because the packet filter would drop all packets except the packets for TCP port 25 destined for the internal network.

Let’s change this a bit. In this case, you create a packet filter for TCP port 25 with the source network ID being the LATDMZ network ID, and the destination network ID being 0.0.0.0 (which represents all network IDs). You select the Drop all packets except those that meet the criteria below option. What happens? All hosts on the DMZ segment are able to send SMTP packets to any network. That’s good, but it would completely break any server publishing rule you have for HTTP, SMTP, NNTP, and so forth because it doesn’t allow outbound access to the high-number ports that the published servers on the DMZ segment need to respond to Internet clients.

What if you selected the Receive all packets except those that meet the criteria below option? In that case, the DMZ interface would allow all traffic through except TCP port 25 traffic from the DMZ to the internal network! You can see how this exception-based packet filtering creates significant constraints on controlling traffic between the DMZ segment and the internal network.

ISA Server DEFCON1

It is possible to achieve a certain level of granular control over traffic moving between the LATDMZ and the internal network. You need to use the allow all traffic except option and then create packet filters for services that you want to prevent from reaching the internal network. For example, some services that you would not want to allow LATDMZ hosts access to on the internal network include HTTP, IMAP4, POP3, SMTP, NNTP, LDAP to Directory Service (TCP 389), LDAP to Global Catalog Server (TCP 3268), Kerberos authentication (UDP and TCP 88), NetBIOS (137/138/139), RPC endpoint mapper (TCP 135), DNS (TCP and UDP 53), and Direct Hosting and NetLogon (TCP 445). Creating packet filters for these ports will block access to the most commonly exploited services on the internal network.
The situation is far from hopeless, however. There are some things that we can do to make the LAT-based DMZ have just about the same functionality and accessibility we have with the private address back-to-back DMZ. What we need to do is identify our requirements, identify the problems, and then come up with solutions.

Requirements for Our LATDMZ Segment
We need to address the following issues with our LAT-based DMZ segment to achieve the same functionality as we have with a private address back-to-back DMZ configuration:

- Prevent LATDMZ hosts from directly accessing the internal network
- Allow Internet traffic into the LATDMZ segment
- Allow internal network hosts access to the Internet
- Allow internal network hosts access to LATDMZ services
- Allow LATDMZ hosts access to the internal network

Let's explore each of these requirements in more detail.

Prevent LATDMZ Hosts from Directly Accessing the Internal Network
The entire reason for creating a DMZ segment is to separate the DMZ and internal network security zones. If you wanted unfettered communication between LATDMZ servers and the internal network, you would just put the servers on the internal network. The entire reason for implementing the DMZ is to protect the internal network in the event a published server is compromised.

The RRAS packet filter we created earlier prevents DMZ hosts from directly accessing the internal network. Unfortunately, this packet filter also prevents internal network clients from directly accessing resources on the LATDMZ segment. This can be a problem because internal network clients often need to access the publicly available servers on the DMZ.

For example, suppose you publish a Web server on a private address back-to-back DMZ. External network clients access the server via the public IP address used by the Incoming Web Requests listener on the external ISA server. Internal network clients access the Web server via the actual private address used by the Web server on the private address DMZ. This prevents the internal network clients from having to “loop back” through the external interface of the external ISA server.

Let's look at another example. Suppose you publish a Web server on your internal network. External network clients access the Web server using the IP address on the external interface of the ISA server that's Web publishing the Web server. Internal network clients access the Web server directly using the Web server’s actual IP address on
the internal network. This prevents the internal network clients from having to “loop
back” through the ISA server to access a server that’s on the internal network.

The problem with the LAT-based DMZ is that the RRAS packet filter prevents
internal network clients from directly communicating with LAT-based DMZ hosts.
We’ll have to come up with a solution if we want the internal network clients to access
resources on the LAT-based DMZ.

Allow Internet Traffic into the LATDMZ Segment
We need to allow Internet traffic into the LATDMZ segment, and the hosts in the
LATDMZ need to be able to respond to Internet hosts generating the traffic. Web and
server publishing rules accomplish this goal. Publishing rules allow inbound traffic into
the LATDMZ. The ISA server automatically creates “dynamic protocol rules” allowing
the LATDMZ host to reply to the requests made by Internet clients.

You don’t need additional RRAS packet filters to allow or deny traffic to and from
the LATDMZ segment. The packet filter we created earlier only affects communica-
tions between the LATDMZ and the internal network. It will have no effect on packets
moving between the LATDMZ and the Internet.

Allow Internal Network Hosts Access to the Internet
Outbound access to the Internet for internal network ISA Server clients is unaffected by
the RRAS packet filter. You control outbound access to the Internet using protocol and
site and content rules. These rules have no effect on how internal network clients access
DMZ hosts. The ISA server sees the LAT-based DMZ as part of the internal network
and therefore doesn’t intervene in communications between trusted network hosts.

Allow Internal Network Hosts Access to LATDMZ Services
Here is where the problems arise. Internal network clients might need to access Web,
SMTP, NNTP, FTP, and other services on the DMZ segment. With the RRAS packet
filter that prevents packets sourcing from the LATDMZ from reaching the internal net-
work, it’s impossible for the internal network clients to directly communicate with
clients on the DMZ segment. The only way the internal network clients will commu-
nicate with the servers on the LATDMZ is via Web and server publishing rules.

The only way internal network clients are going to access services on the
LATDMZ is by “looping back” through the external interface of the ISA server.

I can hear you howling! How many times have we said “Don’t loop back through
the external interface of the ISA server to access resources on the internal network?”
Now it appears that we’re telling you to do just that: “loop back” through the external
interface of the ISA server to access resources on the internal network.
ISA SERVER ALERT

You normally avoid looping back through the external interface of the ISA server by creating a split DNS configuration. The split DNS configuration allows internal network clients to directly access resources on the internal network. The reason why clients loop back through the external interface of the ISA server to reach publishing servers is that the DNS server used by the internal network clients has the IP address on the ISA server used to publish the server. The split DNS configuration uses two DNS zones for the same domain name. One zone is used by the internal network clients, and the second zone is used by external network clients. For more information on the split DNS configuration, check out this author’s article at www.isaserver.org/pages/article.asp?id=995.

However, we’ll plead innocent because you’re not exactly “looping back” through the external interface of the ISA server to access resources on the *internal* network. You’re “looping back” through the external interface of the ISA server to access resources on a LATDMZ. The LATDMZ segment is trusted by the ISA server (because it’s on the LAT), but it’s not trusted by you. You’ve enforced your lack of trust by putting RRAS packet filters in place to prevent direct communications between the internal network and the LATDMZ segment.

You will run into problems with this configuration because the ISA server still sees it as an attempt to “loop back” through the external interface. The reason why we harp on the issue of not “looping back” through the external interface of the ISA server to access internal network resources is that it doesn’t work for SecureNAT clients. If you’ve ever tried to access an internal network server that you’ve published via a server publishing rule from a SecureNAT client on the internal network by looping back through the external interface of the ISA server, you’ve discovered that it won’t work.

SecureNAT clients can’t loop back through the external interface of the ISA server for reasons depicted in Figure 4.17.

When a SecureNAT client sends a request to the ISA server to access resources through the external IP address on the ISA server used in a server publishing rule, the ISA server forwards the request to the publishing server on the LATDMZ. The ISA server does not replace the source IP address of the request that it forwards to the publishing server; therefore, the publishing server on the LATDMZ sees the request coming from the SecureNAT client’s IP address. The publishing server responds directly to the SecureNAT client; it does not respond to the ISA server. The SecureNAT client expects to receive a reply from the ISA server, so the SecureNAT client just drops the response from the publishing server. Of course, with our RRAS packet filters that prevent LATDMZ hosts from directly connecting to internal network hosts, the reply is stopped at the level of the packet filter.
As an example of what happens when you try to loop back through the external interface of the ISA server, let’s look at what happens when we try to connect to an FTP server on the LAT segment from a SecureNAT client on the internal network. We’ll open a command prompt FTP to 192.168.1.33, which is the IP address on the external interface of the ISA server being used to publish an FTP server on the LATDMZ segment. What you’ll see is that the remote host closes the connection.

Figure 4.18 shows a Network Monitor trace obtained at the FTP server on the LATDMZ. Notice the source and destination addresses in the middle frame. The FTP server isn’t responding to the ISA server; it is attempting to respond directly to the SecureNAT client (CLIENTDC with IP address 10.0.0.2). The response fails because our RRAS packet filter prevents any communications between hosts on the LAT-based DMZ segment and the internal network.

One way around this problem is to configure all the clients on the internal network that need to access resources on the LATDMZ to be Firewall clients. Figure 4.19 shows a Network Monitor trace taken at the FTP server of a Firewall client on the internal network connecting to the FTP server on the LATDMZ.
Figure 4.18 Network Monitor Trace at FTP Server

Figure 4.19 Network Monitor Trace Taken at FTP Server of Firewall Client Connecting to It
The FTP server responds to the ISA server, not to the IP address of the Firewall client on the internal network. The Firewall service proxies the requests between the Firewall client and the FTP server on the LATDMZ. In the days of Microsoft Proxy Server 2.0, the Firewall client software was called the "WinSock Proxy" client. When the WinSock Proxy (Firewall service) proxies the requests, it changes the source port so that the server on the LATDMZ responds to the ISA server instead of to the client that made the request.

What if you can’t install the Firewall client on all the computers on the internal network? Maybe you have some Mac- or UNIX-based operating systems on the internal network that need access to publishing servers on the LATDMZ. Is there anything you can do?

Yes, but you’ll have to use another technique to publish servers on the DMZ segment. Instead of using server publishing rules, you’ll have to install the Firewall client on the publishing servers and create a `wspcfg.ini` file. This allows the publishing server to respond directly to the ISA server instead of the SecureNAT client. The Firewall service manages the connection instead of the IP NAT service.

Internal network clients can access HTTP and FTP resources on the LATDMZ if you configure them as Web Proxy clients. The ISA server proxies requests to the publishing FTP server on the LAT network. An interesting finding with the Web Proxy client is you can gain HTTP and FTP access to a LATDMZ server even when only an HTTP publishing rule is active for the publishing server. Figure 4.20 shows a Web Proxy client on the internal network accessing an FTP site on the LATDMZ with only an HTTP publishing rule allowing access to the publishing server through the ISA server.

Another interesting finding is the IP address to which the publishing server responds when the LATDMZ server sends its response to the Web Proxy client. Instead of the external IP address on the ISA server used for the Incoming Web Requests listener, the LATDMZ’s adapter IP address is used. We assume the reason for this is that this IP address is used for the Outgoing Web Requests listener on the LATDMZ segment.

**Allow LATDMZ Hosts Access to the Internal Network**

There are some scenarios where the servers on the LAT segment need to communicate with servers on the internal network. The most common setting for requiring LATDMZ hosts to communicate with internal network clients is when you put an SMTP relay on the LATDMZ. The SMTP relay needs to be able to forward mail for your domains to the SMTP server on the internal network.

We run into the same challenges as we had with internal network clients accessing resources on the DMZ segment. To allow a server on the LATDMZ to access resources on the internal network, the server needs to be a Firewall client, or you will need to publish the internal network server that the LATDMZ needs to connect to by installing the Firewall client on the internal network server and configuring a `wspcfg.ini` file.
What option should you choose? Installing the Firewall client on the server on the LATDMZ requires that you configure an account that the server can use to contact the Firewall service on the ISA server. You really don’t want to extend your internal network’s security zone into the DMZ. You can get around this by configuring the server to use an account that’s local to the ISA server.

However, if the DMZ server is compromised, the intruder might be able to use the credentials used by the Firewall client to gain access to services on the ISA server and through that mechanism access resources on the internal network. There aren’t any documented incidents where this has actually happened, but if you’re in a high-security environment, it’s something to keep in mind.

**ISA SERVER DEFCON 1**

If you install the Firewall client on the LATDMZ server for the purpose of connecting to a publishing server on the internal network, you’ll need to have the user account logged on to the LATDMZ server. The Firewall client needs the logged-on user context to connect to the ISA server using the Firewall client. It’s good security practice to not allow users to be logged on to a publicly accessible server, because an intruder can leverage the logged-on user account to carry out exploits. If the publishing server on the LATDMZ were compromised, the attacker could use the context of that user account to attack the ISA server and possibly internal network hosts.
Chapter 4 • Defense Plan 3: The Internal “Pseudo” DMZ

Making the server on the internal network a Firewall client and configuring a \textit{wspcfg.ini} file creates its own complications. When you publish a server using the Firewall client and a \textit{wspcfg.ini} file, the only way to access that service is by going through the external interface of the ISA server!

For example, look at the scenario in Figure 4.21. The SMTP relay on the LATDMZ segment is published via an SMTP server publishing rule. The SMTP server on the internal network is published via a \textit{wspcfg.ini} file. The SMTP server on the DMZ can access the SMTP server on the internal network by looping back through the external interface of the ISA server. However, hosts on the internal network cannot send SMTP messages to the SMTP server on the internal network! The internal network clients will need to loop back through the external interface of the ISA server to send messages to the SMTP server on the internal network.

\textbf{Figure 4.21 Publishing an FTP Server Using the Firewall Client}

Because of the security implications of putting the Firewall client on a DMZ server, we usually configure the server on the internal network as a Firewall client if we need a server on the DMZ segment to communicate with it when using the private address DMZ configuration.
ISA Server Alert

Using the Firewall client and wspcfg.ini file to publish servers is not for the faint of heart! If you’ve done server publishing using Microsoft Proxy Server 2.0, you’re already acquainted with the methodology and you do it the same way on ISA Server. The key difference is that you do not create server publishing rules and a wspcfg.ini file. You do not need to create server publishing rules when you use the Firewall client on the publishing server. We highly recommend that you review Jim Harrison’s article on the Firewall client at www.isaserver.org/pages/article.asp?id=236. For an example on how to publish an FTP server using a wspcfg.ini file, check out this author’s article on how to publish FTP servers on an alternate port at www.isaserver.org/pages/article.asp?id=196. For an example of how to publish an SMTP server using a wspcfg.ini file, check out www.isaserver.org/pages/article.asp?id=994. To learn about the inability to bind an IP address on the external interface of the ISA server for outbound communications, check out www.isaserver.org/pages/article.asp?id=994.

Using IPSec Policies

RRAS packet filtering allows you to lock down communications leaving the DMZ so that in the event a server on the LATDMZ is compromised, your internal network resources are safe. The main problem with RRAS packet filters is that they are not very flexible. The exception-based policies severely limit granularity of packet filters you can create because you have to take into account both the Internet and the internal network traffic when creating the filters.

If you find that RRAS packet filters are too limiting, IPSec policies might be just what you need to get more granular access control between the LATDMZ and the internal network. Unlike RRAS packet filters, IPSec policies are not based on exceptions. Instead, they are based on IP filter lists, which are collections of packet filters similar to packet filters you configure on the ISA server itself.

Before going into the details of configuring IPSec policies, let’s take a closer look at the components of an IPSec policy.

Elements of an IPSec Policy

An IPSec policy contains these key elements:

- Rules
- Filter lists
- Filter actions
Many people don’t understand or appreciate the power and ease of IPSec policies because the IPSec wizards hide what’s happening when you configure an IPSec policy. You’ll find that when you disable the wizards, you’ll better understand how policies work.

Filter Actions
When a packet matches one of the packet filters in an IP filter list, one of three possible action is carried out after a positive match:

- Permit
- Block
- Negotiate security

You can see the these options in Figure 4.22.

**Figure 4.22 Actions Based on IP Filter List Match**

These three actions are obscured somewhat because they don’t show up right in front of you in the Filter Action tab in the IPSec Rule Properties dialog box (Figure 4.23).

Note in Figure 4.23 that the Deny action is not installed by default; you’ll have to create that yourself. We’ll create a deny filter action later in this chapter.

The most important thing to understand regarding filter actions is they are applied after there’s a match with a particular packet filter in a filter list. The only filter actions are Deny, Permit (Allow), and Negotiate Security.
IP Filter Lists

An IP filter list is a collection of packet filters that IPSec policies use to match incoming and outgoing packets against. When a packet matches the parameters in a particular IP filter list, a filter action is applied to the communication. An IP filter list is just a collection of packet filters that the IPSec policy agent uses to match incoming and outgoing packets against. When there’s a positive match, a filter action associated with that filter list is applied.

Figure 4.24 shows an IP filter list that matches packets sourcing from a machine on the LATDMZ segment to TCP ports 25 and 1433 on IP address 10.0.0.2 on the internal network. Therefore, if this DMZ host sends a packet with any source port to 10.0.0.2 TCP port 25, it will find a match in this IP filter list. In the same way, if this DMZ host sends a packet with any source port to 10.0.0.2 TCP port 1433, it will also find a match in this IP filter list. An IP filter list contains a list of discrete packet filters where only one of them has to be matched. This makes sense, since there is no way a packet with any source port can be sent to TCP port 25 and 1433.

When a packet matches one of the packet filters in the filter list, a filter action is triggered. As stated previously, there are only three filter actions: Permit, Deny, and Negotiate Security.

One thing that makes the IP filter list confusing is that you’re pummeled with multiple filter lists on the IP Filter List tab in the Edit Rule Properties dialog box (Figure 4.25). An IPSec rule only uses a single IP filter list. It gets a bit confusing because you see multiple IP filter lists in the IP Filter Lists frame. Each entry is a list, but you can only select one list per rule. An IPSec policy can contain multiple rules.
IPSec Rules

An IPSec rule contains an IP filter list and a filter action that's applied when a packet matches one of the entries on the filter list. Notice in Figure 4.26 that there are three IP security rules with two of them selected. A single IPSec policy can contain as many rules as you want. One of the active rules denies all traffic to the internal network from the LATDMZ host, and one allows traffic from the LATDMZ host to a specific host on the internal network via TCP 25 and 1433.
This brings up a very important feature of IPSec rules and how they compare to RRAS packet filters. IP rules are applied so that the rule with the most specific match in a filter list is applied.

For example, there are two rules defined in the IPSec policy in Figure 4.26. The Internal TCP 25/1433 rule applies when a packet matches any source port on the LATDMZ computer and the destination is 10.0.0.2 TCP ports 25 and 1433. The Internal Network policy is configured with the source being any port on the LATDMZ computer and the destination being any port on network ID 10.0.0.0/24. Which rule is more specific? Since the first rule is applied to a specific destination computer, it will be applied before the Internal Network rule, which applies to the entire network ID.

The ability to apply more specific rules first gives you many more options than the crude, exception-based rules you must use with RRAS packet filters. You can create a general IPSec rule that denies access to all the network IDs on the internal network. After denying all traffic destined to the internal network from the LATDMZ host, you can then fine-tune the IPSec policy with additional rules that allows you to “punch holes” in the Deny policy you created. As long as the destination IP address or network ID is more specific than the one you created in your general Deny rule, it will be applied before the Deny rule. This will give the LATDMZ host access to particular services and particular servers on the internal network.

Another thing to notice is that you can allow inbound access to the LATDMZ host from all the hosts on the internal network without opening unneeded and unwanted ports from the DMZ to the internal network, and you can allow the ISA server to
control all traffic between the Internet and the DMZ hosts. The IPSec filter lists give you this flexibility.

Let’s look at a scenario and how we would create an IPSec policy to meet our LATDMZ’s requirements.

Creating IPSec Policies to Support Communications between the LATDMZ and the Internal Network

Before creating an IP policy to allow communications between the DMZ segment and the internal network, you need to decide what protocols you want to allow through. First, decide what packets you want to allow from the DMZ host to the internal network, and then decide what packets you want to allow from the internal network to the DMZ.

Before going into the scenario, review the network diagram of the test lab we’re working with in Figure 4.27.

Figure 4.27 Network Diagram

The ISA server has two interfaces. The LATDMZ interface has the IP address 172.16.0.1. There is a server in the LATDMZ with the IP address 172.16.0.2. The ISA
server has an interface on the internal network with the IP address 10.0.0.1, and there is a server on the internal network with the IP address 10.0.0.2.

In this scenario, we want to allow the server in the DMZ to forward SMTP messages to the server on the internal network. We also want to allow internal network clients access to HTTP and FTP on the DMZ server. All other packets between the internal network and the LAT-based DMZ segment are dropped.

Perform the following steps to create the IPSec policy:

1. On the DMZ server, click **Start**, point to **Programs**, and point to **Administrative Tools**. Click on **Local Security Policy**.
2. In the **Local Security Policy** console, right-click on the **IP Security Policies on Local Machine** node and click the **Create IP Security Policy** command.
3. Click **Next** on the **Welcome to the IP Security Policy Wizard** page.
4. On the **IP Security Policy Name** page, type **DMZ Host** in the **Name** text box and click **Next**.
5. On the **Requests for Secure Communication** page, remove the check mark from the **Activate the default response rule** check box. We want to use an IPSec policy to provide packet filtering, not to negotiate security. This is the best-kept secret of IPSec policies! You can use them for advanced packet filtering and you don’t have to worry about negotiating security. Click **Next**.
6. This brings you to the **Default Response Rule Authentication Method** page. Since we’re not negotiating security and we’re not using the default response rule, it doesn’t matter what authentication method we choose. However, if you choose the **Windows 2000 default (Kerberos V5 protocol)** option, you’ll get an error message. Select the **Use this string to protect the key exchange (preshared key)** and type **987654**. Click **Next**.
7. On the **Completing the IP Security Policy Wizard** page, leave the check mark in the **Edit Properties** check box. This brings up the dialog box that allows you to create a rule that includes a filter list and filter action. Click **Finish**.
8. In the **DMZ Host Properties** dialog box, remove the check mark from the **<Dynamic>** IP filter list. Remove the check mark from the **Use Add Wizard**. Click **Add** so you can create the filter list.
9. The **IP Filter List** tab (Figure 4.28) appears first in the **New Rule Properties** dialog box. The first thing we need to do is create a filter list to match packets against. Click **Add** to begin creating the filter list.
10. In the **IP Filter List** dialog box, type **All IP Traffic** in the **Name** text box (Figure 4.29). Remove the check mark from the **Use Add Wizard** check box. Click **Add** to create a filter.

**Figure 4.29 The IP Filter List Dialog Box**

11. In the **Filter Properties** dialog box (Figure 4.30), make sure the **Source address** is set for **My IP address**. For the **Destination address**, select the **A specific IP subnet** option. In the **IP Address** text box, type **10.0.0.0**, and in...
the Subnet mask text box, type 255.255.255.0. Leave the check mark in the Mirrored check box. This will apply the filter to packets with the exact opposite source and destination address and port numbers.

**Figure 4.30 The Filter Properties Dialog Box**

12. Click on the Protocol tab (Figure 4.31). Note that the default setting matches all protocols—this is what we want. This filter matches all packets, regardless of protocol. We’ll then associate a Deny filter action to this filter. After creating this filter, we’ll poke holes in it with another IPSec rule to include in this policy. Click Apply, and then click OK.

**Figure 4.31 The Filter Properties Dialog Box**
13. Click **Close** in the **IP Filter List** dialog box. Notice there is already an **All IP Traffic** filter. Double-click on that filter to read the description (Figure 4.32). The dialog box informs you that even though you created a filter that matches all IP traffic, it will *not* match broadcast, multicast, Kerberos, RSVP, and IKE packets. The implication is that you won’t be able to use an IPSec policy to block these types of packets. This should not present too much of a problem, as these protocols are not typically used to exploit servers on the internal network. Click **Cancel** in the **IP Filter List** dialog box.

**Figure 4.32 The IP Filter List Dialog Box**

14. Click the option button to the left of the **All IP Traffic** filter list.

15. Click the **Filter Action** tab (Figure 4.33). Note that there is a Permit filter and two filters that are used to negotiate security. The Permit filter allows traffic-matching filters in the filter list you selected in the **IP Filter List** tab. In our scenario, we want to begin by denying all traffic between the DMZ and the internal network. In order to do so, we’ll have to create a new filter action. Remove the check mark from the **Use Add Wizard** and click **Add** to create the new filter action.

16. In the **New Filter Action Properties** dialog box (Figure 4.34), select the **Block** option. Click on the **General** tab. Type **Block** in the **Name** text box. Click **Apply**, and then click **OK**.

17. The Block filter action appears in the **Filter Action** dialog box (Figure 4.35). Select that option by clicking on the option button to the left of the **Block** filter action. This completes the configuration of the rule. What this rule does is match all packets for all protocols and applies the Block action. If you did nothing else, no traffic would be allowed between this machine and the internal network. Click **Apply**, and then click **OK**.
You now have one rule that’s active on the **Rules** tab of the **DMZ Host Properties** dialog box. Remember that you can have multiple rules active in a single IPSec policy. You only need to create a single rule per filter action. If you wanted to deny access to more subnets or individual IP addresses, you could add more filter lists to the All IP Traffic rule.
The next rule we need to create will contain filter lists tied to a Permit filter action.

1. In the DMZ Host Properties dialog box, click Add to add another rule.

2. In the New Rule Properties dialog box, click Add on the IP Filter List tab to add a new filter list.

3. In the IP Filter List dialog box, type DMZ-Internal in the Name text box. Click Add to add a filter.

4. In the Filter Properties dialog box (Figure 4.36), make sure the My IP Address option is selected for the Source address. For the Destination address, select the A specific IP Address option. In the IP address text box, type 10.0.0.2 (which is the IP address of the SMTP server on the internal network). Leave the check mark in the Mirrored check box. This will allow the server on the internal network to reply to the DMZ server.

5. Click on the Protocol tab (Figure 4.37). In the Select a protocol type drop-down list box, select the TCP option. In the Set the IP protocol port frame, select the From any port option and the To this port option. In the To this port text box, type 25. Click Apply, and then click OK.

6. In the IP Filter List dialog box click Add to add a filter.
7. In the Filter Properties dialog box, make sure the A specific IP Subnet option is selected for the Source address. In the IP Address text box, type 10.0.0.0 in the IP Address text box. Type 255.255.255.0 in the Subnet mask text box. For the Destination address, select the My IP Address option. Make sure the Mirrored option is selected, as this will allow the server on the internal network to reply to the DMZ server. Click on the Protocol tab. In the Select a protocol type drop-down list box, select the TCP option. In the Set the IP protocol port frame, select the From any
port option and the To this port option. In the To this port text box, type 80. Click Apply, and then click OK.

8. Repeat the procedure in step #7, but use the destination port 21.

9. In the IP Filter List dialog box, click Add to add a filter. In the Filter Properties dialog box, make sure the My IP Address option is selected for the Source address. For the Destination address, select the A specific IP Subnet option. In the IP address text box, type 10.0.0.0 (which is the IP address of the SMTP server on the internal network). Leave the check mark in the Mirrored check box. This will allow the server on the internal network to reply to the DMZ server. Click on the Protocol tab. In the Select a protocol type drop-down list box, select the TCP option. In the Set the IP protocol port frame, select the From this port option and the To any port option. In the From this port text box, type 20. Click Apply, and then click OK.

10. Your IP Filter List should look something like what appears in Figure 4.38. Click Close in the IP Filter List dialog box.

Figure 4.38 The IP Filter List Dialog Box

11. In the New Rule Properties dialog box, select the DMZ-Internal filter list by clicking the option button to the left of it. Click on the Filter Action tab and select the Permit option. Click Apply, and then click OK.

12. Click Close in the DMZ Host Properties dialog box.

13. Right-click on the DMZ Host Policy and click the Assign command.

14. Go to the Administrative Tools menu and click the Services entry. In the Services console, find the IPSEC Policy Agent entry and click the Restart button in the console’s button bar.
You can test this policy by going to a host on the internal network and using the command-line FTP application to connect to the FTP server on the DMZ segment. You should be able to connect with no problem. Be sure to use PORT mode FTP. The filter list you created to allow traffic through does not support PASV mode FTP connections.

You should also try using the browser on a host on the internal network and connect to the Web server on the DMZ segment. You’ll have no problems connecting! However, if you try to telnet to TCP port 25 on the DMZ host, it won’t work because the filter list for the Permit rule doesn’t include an entry that allows hosts on the internal network to connect to TCP port 25 from any source port number.

Now that you see the logic behind creating IP policies, you will be able to create policies to meet your own requirements. Keep in mind the following facts:

- You can activate only one policy at a time.
- An IPSec policy can contain multiple rules.
- You usually will need to create one rule for each filter action.
- Each rule can contain multiple filter lists.
- A filter list is a collection of packet filters that are used to compare against packets coming to and leaving the interface; if there’s a match, the filter action associated with the rule is applied.
- A single filter action is applied to a rule.
- You cannot create a Block filter action; there is no default Block filter action included with Windows 2000.
- Always disable the default response rule; enable only rules you intend to use.
Summary

In the chapter, we went over techniques that you can use to implement a private address, LAT-based DMZ segment. The ability to use an internal network segment for a DMZ is important because you should be able to take advantage of the Web and server publishing rules to make servers available on the DMZ. The only other option for a trihomed ISA server for publishing servers on a DMZ segment is to use public IP addresses on the DMZ segment, and then use packet filters to make those servers available to the Internet. As you learned in the last two chapters, packet filters are not very flexible and do not leverage the security features provided by Web and server publishing rules.

You can use TCP/IP security to perform basic access control for incoming packets to the DMZ host. TCP/IP filters don’t do much to protect the internal network from a DMZ host, but they can prevent the DMZ host from packets arriving on hosts on the internal network. TCP security filters only apply to incoming packets and have no influence on packets leaving the host configured to use TCP/IP security filtering. One advantage of using TCP/IP security filters is that they work in kernel mode and thus are less subject to compromise.

RRAS packet filters provide a powerful and effective method to control traffic moving between the LAT-based DMZ segment and the internal network. RRAS packet filters work on an exception basis; you can allow all traffic except for that for which you create filters, or you can deny all traffic except for that for which you create filters. While you can obtain a certain level of granularity over packets moving into and out of the DMZ segment, the exception-based filter makes it difficult to create a universal Deny rule with exceptions. RRAS packet filters depend on the Routing and Remote Access Service (RRAS), which runs as a user-mode process. The operating system can continue to run even if there is an access violation in the RRAS service, which can create a security risk to the internal network if the RRAS service is disabled on the ISA server.

IPSec policies are implemented on the DMZ host itself. IPSec policies leverage sophisticated filter lists and filter actions that comprise IPSec rules. You can create a general, global Deny rule and then create exceptions to the global Deny by creating more specific rules. The IPSec policy agent is a user-mode process, and thus can be interrupted and the operating system will still run, which can cause a security risk to internal network clients if the agent is compromise on the DMZ host.
Defensive Tactics Fast Track

Using TCP/IP Filtering (TCP/IP Security)

- TCP/IP filtering controls traffic sent to the interface of the computer on which TCP/IP filtering is configured.
- TCP/IP filtering has no effect on traffic leaving the interface.
- TCP/IP filtering has no effect on ICMP traffic.
- You can control inbound UDP and TCP traffic using TCP/IP filtering.
- You can use TCP/IP filtering to control IP protocol type traffic (such as GRE IP Protocol 47).
- TCP/IP filtering is used mainly to protect DMZ hosts from traffic originating from internal network hosts.
- TCP/IP filtering runs in kernel mode, which confers it a higher level of security.

Using Routing and Remote Access Service (RRAS) Packet Filters

- RRAS packet filters can control traffic moving between any of the interfaces on the ISA server computer.
- You can use RRAS packet filters and not run a VPN server or VPN gateway on the ISA server; the only thing you need to do is disable the packet filters that the ISA Server VPN wizard creates.
- RRAS packet filters work on an exception basis; you can allow all traffic except for that which matches your packet filters, or you can deny all traffic except for that which matches your packet filters.
- Exception-based filtering makes it difficult to control traffic on a granular basis because you cannot apply filters to all networks without adversely affecting Web and server publishing scenarios on the DMZ segment.
- The most effective RRAS packet filters are those that prevent traffic sourcing from the LATDMZ segment from reaching any host on the internal network.
- You can create complex filters denying specific traffic from the LATDMZ into the internal network. For example, you can select the allow all traffic
except option and then create packet filters for NetBIOS and SMB/CIFS ports, SMTP, NNTP, Kerberos, IPSec, POP3, Telnet, or whatever services you might be running on the internal network that you do not want LATDMZ hosts to access.

- RRAS packet filters can be used together with IPSec policies to create a multilayer security configuration.

### Using IPSec Policies

- IPSec policies are based on IPSec rules, filter lists, and filter actions.
- You should create one IPSec rule per filter action.
- You can put multiple packet filters into a single filter list.
- You can use multiple IPSec rules in an IPSec policy.
- IPSec policies allow a more granular level of access control than RRAS policies do.
- IPSec policies apply rules based on more specific filter lists preferentially over more general filter lists; filter actions associated with a specific IP address or subnet will be applied before a filter action that applies to all computers.
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 and click on the “Ask the Author” form.

Q: Why would I want to create a LAT-based DMZ?
A: There are a number of reasons to create a LAT-based DMZ. The most important is so you can take advantage of the features provided by the Firewall and Web Proxy services to filter traffic as it moves from the Internet to the published servers on the DMZ segment. If you don’t use the LAT-based DMZ configuration, you’ll have to implement a back-to-back DMZ—which introduces higher cost and high complexity. If you don’t want to use a back-to-back DMZ, your only other “official” choice is the public address DMZ segment. When you use public addresses, you have to use ISA Server packet filters to move traffic between the Internet and the DMZ segment. By using the LAT-based DMZ configuration, you get around the limitations of the traditional trihomed DMZ configuration and take advantage of ISA Server’s Web and server publishing rules to publish servers on a DMZ segment.

Q: I don’t want to allow any traffic to move between the LAT-based DMZ segment and the internal network. I want my DMZ to be a true DMZ that is not trusted by the internal network. How do I prevent direct communications between the LAT-based DMZ segment and the internal network?
A: The best way to enforce this type of demarcation between the LAT-based DMZ and the internal network is by using RRAS packet filters. RRAS packet filters are configured on the ISA Server machine and prevent traffic sourcing from the DMZ segment from reaching the internal network. Because these filters are applied on an exception basis, the easiest way to do this is to prevent all traffic from the DMZ network ID from being passed directly to the internal network ID.

Q: I find the RRAS packet filters too restrictive. What I want to do is allow only selected traffic from the DMZ segment to the internal network, and then allow all traffic from the internal network directly to the DMZ. The reason I want to do this is to take some of the heat off the ISA server and let the internal network clients directly communicate with the servers on the DMZ. How can I do this?
A: The best way to accomplish this is by using IPSec policies. IPSec policies use filter lists, which are a collection of packet filters that are used to match against a filter action. You can create a global Deny rule that prevents a DMZ host from sending any traffic to the internal network. Then, you can create most specific rules that contain filters that allow specific protocols access to specific servers on the internal network. This allows hosts on the LAT-based DMZ segment to send and receive traffic from specific servers on the internal network.

Q: How does TCP/IP Security fit into the LAT-based DMZ configuration?

A: TCP/IP Security can be used to control what protocols are allowed inbound access to a server on the LATDMZ segment. TCP/IP Security filters are more focused on protecting LAT hosts from clients on the internal network. The LATDMZ servers are protected from Internet hosts by the ISA server. The ISA server controls what protocols are allowed inbound from the Internet to the hosts on the DMZ segment. TCP/IP Security can be used to block all traffic to a LATDMZ server except for the protocols you want to allow. You can accomplish the same goal using RRAS packet filters and IPSec policies. The advantage of TCP/IP Security filters is that they run in kernel mode, as compared to the RRAS packet filters and IPSec policies, which depend on user-mode processes.

Q: I want to run a front-end/back-end Exchange Server configuration and put the front end in the LATDMZ segment. What can I do to make this happen?

A: The first question you have to ask is, “Why should I put the front end in the DMZ segment?” If you’re doing it for security reasons, think about the security implications of extending the internal network security zone into the DMZ segment. The entire purpose of the DMZ is to create a completely separate and distinct security zone. By creating such a separate security zone, even if a server on the DMZ is compromised, the accounts on the internal network are secure. The intruder will not be able to leverage accounts on the compromised server to access resources on the internal network. If you put a front-end server on the DMZ segment, you must make that front-end server a member of an internal network domain.