Novell BorderManager®

3.9

April 5, 2007

www.novell.com

PROXY AND FIREWALL OVERVIEW AND PLANNING GUIDE

Novell®

Legal Notices

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

Third-Party Materials

Contents

About This Guide

This documentation presents an overview of the Novell® BorderManager® 3.9 components and provides the background information you need to plan your implementation of this software. It includes the following sections:

- Chapter 1, "Overview," on page 11
- Chapter 2, "Packet Filtering Overview and Planning," on page 29
- Chapter 3, "Access Control Overview and Planning," on page 41
- Chapter 4, "NAT Overview and Planning," on page 53
- Chapter 5, "Proxy Services Overview and Planning," on page 63
- Chapter 6, "Novell BorderManager Planning Scenarios," on page 93

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *Novell BorderManager 3.9 Proxy and Firewall Overview and Planning Guide*, visit the Novell Documentation Site (http://www.novell.com/documentation/nbm39/index.html).

Additional Documentation

This Proxy and Firewall Overview Guide is a part of documentation set for Novell BorderManager 3.9. The other documents include:

- *Novell BorderManager 3.9 Installation Guide*
- *Novell BorderManager 3.9 Administration Guide*
- *Novell BorderManager 3.9 Virtual Private Network Client Installation Guide*
- *Novell BorderManager 3.9 Troubleshooting Guide*
- *Novell BorderManager 3.9 Virtual Private Network Deployment Frequently Asked Questions*

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

Also, a trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux or UNIX, should use forward slashes as required by your software.

1 Overview

This section provides an overview of the Novell® BorderManager® services that you can use to successfully manage your network borders. It also discusses the requirements for managing and controlling access to a network border. Subsequent sections provide a more detailed description of each Novell BorderManager service.

NOTE: For procedural information, see *Novell BorderManager 3.9 Installation Guide*.

This section contains the following:

- Section 1.1, "Novell BorderManager Features," on page 11
- Section 1.2, "Implementing Network Border Security," on page 14
- Section 1.3, "Firewall Technologies," on page 16
- Section 1.4, "Improving Network Performance," on page 25
- Section 1.5, "Managing Novell BorderManager Services," on page 27
- Section 1.6, "Where to Go from Here," on page 28

1.1 Novell BorderManager Features

When you connect your private network to the public Internet, security at the border and network performance become key issues, as is protecting data on your intranet.

Novell BorderManager enables you to manage and protect the border where networks meet. Although the border most commonly referred to is the one between the corporate intranet and the Internet, the borders between segments of a company intranet also must be managed and protected. Because Novell BorderManager is specifically designed to address and solve the most critical issues involved with managing a network border, the administrator's job becomes infinitely easier when this product is deployed throughout the network.

The Novell BorderManager software provides a secure connection from a corporate intranet to the Internet. Novell BorderManager runs on a NetWare® 6.5 SP 6 or OES SP 2 operating system and uses iManager for server configuration. Server configurations are stored in the NDS® or the Novell eDirectory™ database. NDS or eDirectory enables you to control user and user group access to the World Wide Web.

The task of managing a network border is not a simple one, but it can be more easily understood when broken down into the solution categories provided by Novell BorderManager and briefly described in the next section. This section covers:

- Section 1.1.1, "Firewall Solutions," on page 12
- Section 1.1.2, "Network Border Considerations," on page 12

1.1.1 Firewall Solutions

Novell BorderManager offers comprehensive, effective firewall solutions that include the following technologies:

- **Packet Filtering:** Packet filters provide Network-layer security to control the types of information sent between networks and hosts. Novell BorderManager supports Routing Information Protocol (RIP) filters and packet forwarding filters to control the service and route information for the common protocol suites, including Internetwork Packet Exchange™ (IPX™) software and TCP/IP.
- **Proxy Services:** This component uses caching to accelerate Internet performance and optimize WAN bandwidth use. Proxy Services also allows protocol filtering and improves security by hiding private network domain names and addresses, and sending all requests through a single gateway.
- **Access Control:** Access control is the process by which user access to Internet and intranet services is regulated and monitored. Specifically, the BorderManager access control software allows or denies access requests made through the Novell Proxy Services.
- **Network Address Translation (NAT):** NAT allows IP clients on your local network to access the Internet without requiring you to assign globally unique IP addresses to each system. In addition, NAT acts as a filter, allowing only certain outbound connections and guaranteeing that inbound connections cannot be initiated from the public network.
- **Virtual Private Network (VPN):** A VPN is used to transfer sensitive information across the Internet in a secure fashion by encapsulating and encrypting the data. A VPN can also be deployed in intranets where data security is required between departments.
- **Novell BorderManager Alert:** The Novell BorderManager Alert monitors server performance and security, and reports potential or existing server problems that affect the performance of configured Novell BorderManager services.
- **Novell BorderManager Authentication Services:** Authentication services enable remote users to dial in to NetWare networks and access network information and resources.

1.1.2 Network Border Considerations

You must consider the following when establishing and maintaining control over your network borders:

- **Security:** You need to protect the intranet from security breaches, as well as preventing unauthorized access to the Internet.
- **Performance:** Performance is a critical issue if access to the Internet and intranet is to be useful. You must be able to optimize Internet and intranet access, even over slow dial-up lines.
- **Management:** You need to establish security over all Internet and intranet access points by involving intranet as well as Internet access security. You need a centralized way to manage all Internet and intranet access points.
- **Secure Remote Connectivity:** You need to send information in a cost-effective and secure way.

You must also consider the setup and ongoing maintenance costs of your Internet access points when you establish and maintain control over your network borders. Novell BorderManager addresses all these network management and protection considerations.

Security

Security is one of the major considerations when connecting a corporate intranet to the Internet. Protecting information and systems from unauthorized access can be just as important when considering a network segment located within the company intranet. Keep in mind that some of the most knowledgeable software experts might also be your employees. More than 80 percent of data is stolen internally. Creating a security mechanism to guard your network border is commonly referred to as creating a firewall.

Novell BorderManager provides the following security features that you can use to create a network firewall:

- Packet filtering
- Network Address Translation (NAT)
- Application proxies
- Access control
- SurfControl*

For more information on security, firewalls, and the Novell BorderManager security features, refer to Section 1.2, “Implementing Network Border Security,” on page 14.

Performance

With the emergence of the concept of Internet time, companies understand the urgency of speeding up access to and from the Internet. Many companies rely on the Internet to exchange products and information with colleagues and customers.

The demand to speed up access to information from within a company is just as strong. Many companies have decided to move all corporate information, including documents, forms, and procedures, to an intranet Web site. If all company information is located on a Web site, and employees are working on company time, it is essential that employees be able to access and gather this data quickly.

Novell BorderManager provides the following performance features:

- Forward acceleration, or standard proxy caching
- Reverse acceleration, or Web server acceleration
- Hierarchical caching

For more information on these Novell BorderManager performance features, refer to Section 1.4, “Improving Network Performance,” on page 25.

Management

Consistently managing network borders can be difficult when each border must be managed separately. This task is further complicated if each border uses different routing hardware and software. Novell BorderManager eases this problem by enabling the administrator to manage Novell BorderManager servers from a centralized location.

For more information on managing Novell BorderManager, refer to Section 1.5, “Managing Novell BorderManager Services,” on page 27.

Secure Remote Connectivity

With the increasing need to access and send information online, it is essential to have a cost-effective and secure method for transmitting the information. In the past, many companies chose to build private networks using leased dial lines, but this approach can be expensive. Today, it might make more sense for a company to use the Internet to send and receive secure online information. The Novell BorderManager Virtual Private Network (VPN) enables you to use the Internet to send and receive information securely using an encrypted data stream between hosts and clients.

For more information on the Novell BorderManager VPN, refer to "Virtual Private Networks" on page 22.

1.2 Implementing Network Border Security

The Internet and the company intranet are both insecure environments. When an organization connects to the Internet or creates an internal intranet with segments that maintain confidential information, it puts its internal information at risk. Both situations call for secure, controlled access and increased security to protect valuable corporate information. The first step in implementing network security is to establish a security policy.

This section contains the following subsections:

- Section 1.2.1, "Security Policy Guidelines," on page 14
- Section 1.2.2, "Deciding What to Include in a Security Policy," on page 15
- Section 1.2.3, "Creating a Security Policy," on page 15
- Section 1.2.4, "Establishing a Security Policy Using Novell BorderManager," on page 16

1.2.1 Security Policy Guidelines

Using Novell BorderManager to secure your network borders is just one of the several steps required to implement network security at your company. Implementing a secure network takes a great deal of planning and cooperation from the employees at your company. To successfully implement a secure intranet, you must create a security policy. Creating a security policy can be a long, complex process, but it is essential to the success of creating a secure network. Although technology cannot guarantee a completely secure system, you can take steps to prevent the misuse of data and systems in your organization.

A security policy should be a guideline for all employees and administrators in your organization. It should consist of a set of rules expressing the goals you want to meet in securing and controlling access to your networks. The policy you implement depends on the technologies available to you to carry out the rules you establish.

Follow these general guidelines when writing your security policy:

- Explain why the policy was created. This is useful when you need to make changes later, and need to recall why certain rules were set up.
- Use plain language. This helps employees read and understand the policy.
- Detail the responsibilities of the employees and administrators. For example, spell out that employees must keep their passwords secret.
- Assign authority. Delegate responsibility when security breaches occur and the policy is not being followed. Include any punitive actions that can result, including reprimands or dismissal.

1.2.2 Deciding What to Include in a Security Policy

Consider the following general issues when deciding what to include in your security policy. You might need to add to this list, based on conversations with staff and administrators in your organization. Your security policy should include rules for the following:

- Assigning and accessing accounts
- Connecting objects to your network, including connecting a host or client
- Connecting to the Internet
- Protecting sensitive information on intranet Web or FTP servers
- Publishing information on the Internet
- Connecting remote users, sites, and customers to your network
- Using e-mail
- Protecting company-confidential information
- Recovering from security breaches
- Enforcing rules for multiple sites, and creating a consistent policy among sites for easier maintenance

IMPORTANT: The preceding list is not comprehensive; rather, it gives you a general idea of the issues you need to address and provides a starting point for your security policy. Refer to the available information on Internet security, both online and in bookstores, for more details about designing a complete and comprehensive security policy for your network.

1.2.3 Creating a Security Policy

1. Research potential security policies using sources available on the Internet, as well as published material.
2. Determine the following information about your organization:
    - Types of applications and data: Identify categories and determine what needs to be protected and what can be made public, both within and outside the company.
    - Current relationships: Determine, for example, whether you want to support customer and supplier access.
    - Employees who need access to information: Categorize this further by determining who needs access to what information.
3. Determine how the policy can be changed in the future. Specify how new technology and requirements will be incorporated into your organization and the security policy.
4. Analyze the security policy with regard to risk and cost. This process can become very analytical and might be better accomplished by hiring a consultant.
5. Publish the security policy. Make sure that all employees read and understand both the policy and their responsibilities.
6. Implement the policy. This involves implementing the firewall and enforcing the guidelines established by the security policy.
7. Enforce the policy. The policy is useless if you do not make sure it is adhered to by all concerned.

8. Review and update the security policy on an ongoing basis to deal with new issues and changes to the network.

IMPORTANT: This information is a guide only and is not meant to provide all the data you need to create a corporate security policy. For more information on network security and implementing a security policy, read one of the many third-party publications that provide detailed information on this subject.

1.2.4 Establishing a Security Policy Using Novell BorderManager

You can control access to a Novell BorderManager security on your network by implementing the following rules:

- When installing or upgrading Novell BorderManager, disconnect the server from the public network.
- Control network access to a Novell BorderManager server as follows:
  - Do not configure host utilities such as RCONSOLE or XCONSOLE that provide remote access to the system.
  - Do not use a Novell BorderManager server to support data hosting applications such as file and print services.
  - Restrict Simple Network Management Protocol (SNMP) access to the system.
  - Change the default SNMP community string.
  - Control NetWare Core Protocol™ (NCP™) connections to the system by setting packet signatures to the highest level (level 3).
  - Block source address spoofing by applying packet filters to public interfaces.
  - Restrict physical access to the server.
- Scan network devices and workstations for viruses.
- Establish a 7-day, 24-hour emergency procedure for handling security breaches.
- Disconnect the Novell BorderManager server from the public network if a security breach is suspected.
- Encourage users to log out of the network and lock their workstations at the end of the day.
- Mandate periodic changes to passwords. Discourage users from choosing personal information, such as names or birth dates, when setting new passwords.
- Reference RFC 1244 to formulate guidelines and further implement a site security policy.

1.3 Firewall Technologies

This section discusses several types of firewalls and describes the firewall services provided by Novell BorderManager. It contains the following subsections:

Firewalls are a combination of hardware and software that reduce the risk of a security breach into a private intranet. An effective firewall between the intranet or private network and the Internet, or between intranet segments, enforces corporate security and access control policies. A firewall also helps regulate the type of traffic that can access the intranet and provides information about that traffic to the administrator.

You can set up your firewall to deny access to a private network from the Internet, but to allow access to the Internet. Or you can allow some access from the Internet, but only to selected servers for e-mail or general corporate information.

The purpose of a firewall is to create a system that prevents unauthorized users from accessing proprietary information. As previously mentioned, designing an effective security policy that meets your needs requires careful planning and consideration of your objectives. This section focuses on understanding the firewall portion only.

The Open System Interconnection (OSI) model shown in the following table provides a view of each layer mapped to the corresponding Internet firewall technologies. Some technologies span more than one layer. Higher levels in the OSI model provide a better or more detailed capability of controlling data that enters your network, at the expense of performance. Lower levels require less time to route data but sacrifice security for performance.

Table 1-1 *Open System Interconnection (OSI) Model*

OSI Layer	Firewall Technology
Application	Virtual Private Network (VPN) Internet Object Caching
Presentation	VPN
Session	VPN
Transport	VPN
	IPX/IP
Network	VPN
	Network Address Translation (NAT)
	Packet filtering
Data Link	VPN
	Point-to-Point Protocol (PPP)
	Packet filtering
Physical	Not applicable

1.3.1 Designing a Firewall

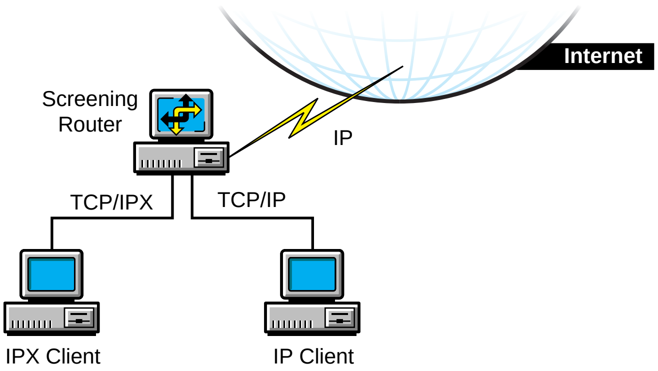
Although a firewall is sometimes referred to as a single technology, it is actually a combination of several services that work together as a protective layer to ensure a secure network border. These technologies build on the basic security already available in many Internet services. Firewalls provide security for services that do not have security, for example, e-mail. Firewalls also protect hosts. There are several basic types of firewalls:

- "Screening Routers" on page 18
- "Bastion Hosts" on page 18
- "Dual-Homed Hosts" on page 19
- "Screened Hosts" on page 19
- "Screened Subnets" on page 20
- "Tri-Homed Hosts" on page 21

Screening Routers

A screening router is the most basic type of firewall and uses only the packet filtering capability to control and monitor network traffic that passes though the border. Screening routers on a server with packet filtering can block traffic between networks or, for example, traffic to or from specific hosts on an IP port level. For example, you can let employees on your intranet use Telnet, but you can bar any Telnet activity from the Internet. Direct communication is usually permitted between multiple hosts on the private network and the Internet. The following figure shows a basic example of how a screening router works.

Figure 1-1 *Firewall Using Screening Routers*



The risk of break-in is large with this type of firewall because each host on the private network is exposed to the Internet and is still a potential break-in point. Unauthorized users can detect and use internal addresses to access information within the firewall. To avoid this, screening routers can be set to look at the source address of each incoming IP header instead of the destination address, and drop private addresses that come from the Internet.

Bastion Hosts

A bastion host represents the private network on the Internet. The host is the point of contact for incoming traffic from the Internet, and as a proxy server allows intranet clients access to external services.

A bastion host runs only a few services, such as, e-mail, FTP, Domain Name System (DNS), or Web services. Internet users must use the bastion host to access a service. A bastion host does not require any authentication or store any company-sensitive data.

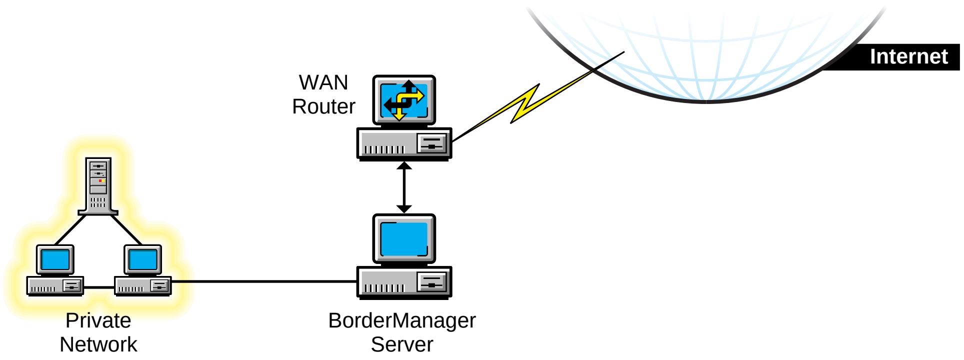
Dual-Homed Hosts

A dual-homed host is based on a server with at least two network interfaces. The host acts as a router between the network and the interfaces to which it is attached. To implement a dual-homed host type of firewall, the routing function is disabled. Therefore, an IP packet from one network (for example, the Internet) is not routed directly to the other network (for example, the intranet). Systems inside and outside the firewall can communicate with the dual-homed host but cannot communicate directly with each other.

A dual-homed host blocks direct traffic between the private (protected) network and the Internet. The following figure provides an example of a configuration in which a WAN router provides general WAN connectivity, packet filtering, and access to the Novell BorderManager server. Private network users can access the Internet by using Proxy Services which are running on the Novell BorderManager server.

The router allows traffic only to and from the Novell BorderManager server. Break-in is limited to other hosts reachable from the Internet, although any illegal access severely compromises security.

Figure 1-2 *Dual-Homed Host Firewall*



Screened Hosts

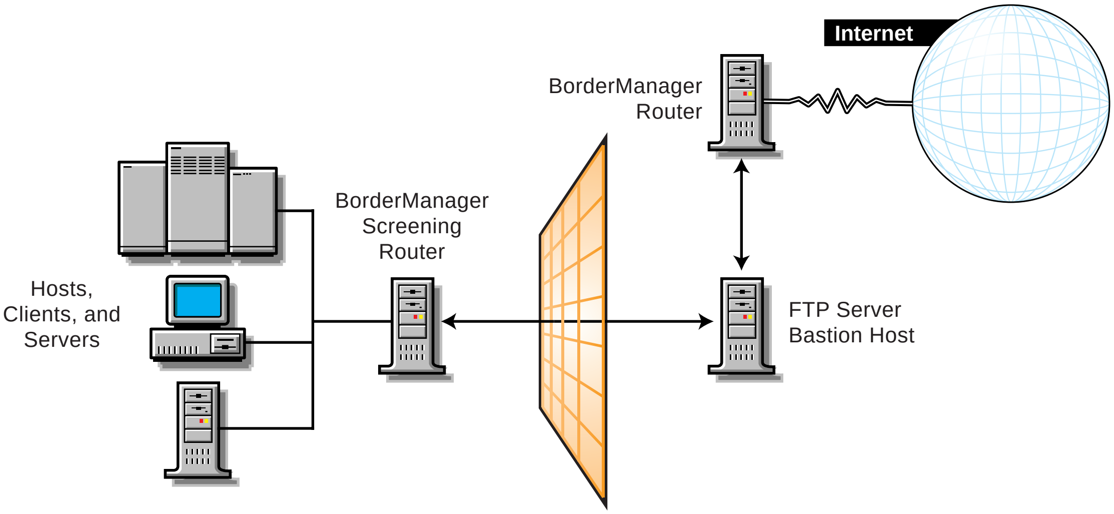
A screened host uses a combination of a bastion host and a screening router, as shown in the following figure. The screening router adds security by providing Internet access to deny or permit certain traffic from the bastion host. It is the first stop for traffic, which can continue only if the screening router lets it through.

For additional security, you could set up a bastion host for each type of service: HTTP, FTP, and e-mail. The screening router then sends the corresponding traffic to the appropriate bastion host.

In this example, the Novell BorderManager server and the WAN router can be reached from the Internet. In addition, the Novell BorderManager server acts as a screening router on the private network. Using Network Address Translation (NAT) and packet filtering, the Novell BorderManager server can be configured to block traffic on specific ports, and only a select number of services can communicate with it.

This type of firewall is fairly secure because security risk is limited to the Novell BorderManager server.

Figure 1-3 *Screened Host Firewall*

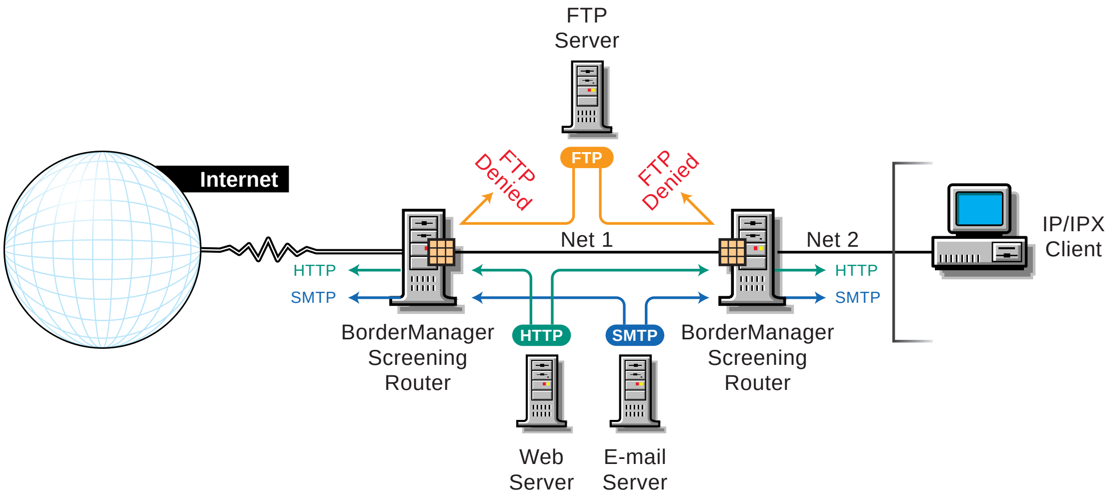


Screened Subnets

The screened subnet is a variation of a screened host. In screened subnetting, the bastion host is placed on its own subnetwork. Two screening routers are used to do this: one between the subnet and the private network (with the bastion host) and the other between the subnet and the Internet. The first screening router between the private network and the screened subnet denies all services from crossing into the subnet. The screened subnet allows only specified services. An example of a screened subnet firewall is shown in the following figure.

In this configuration, the Novell BorderManager server is used as a proxy server. Both IP routing and IP forwarding are disabled to prevent any direct access between the private network and the public Internet.

Figure 1-4 *Screened Subnet Firewall*

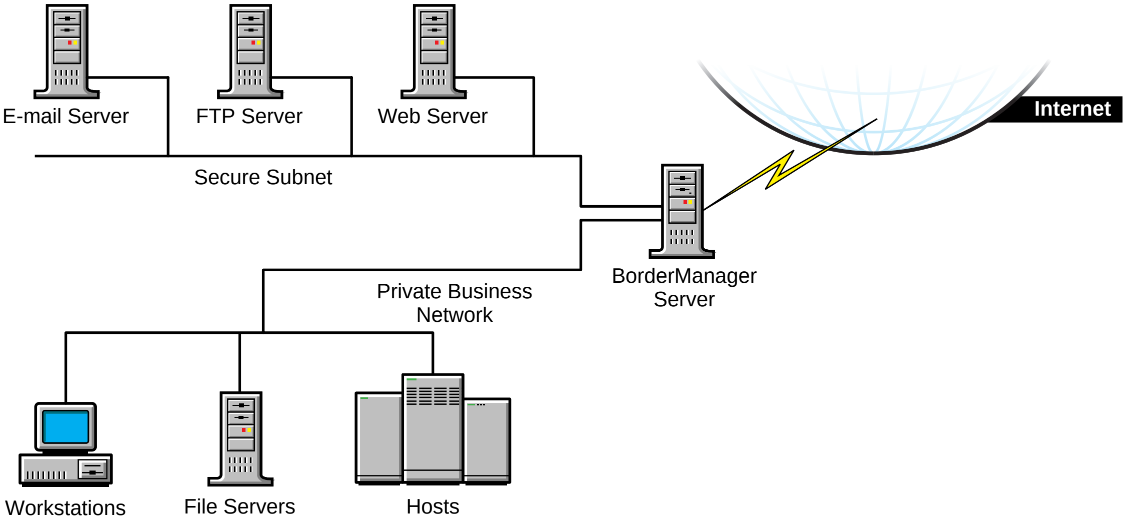


IMPORTANT: Although a firewall focuses on the restriction and use of internetwork services, for complete protection you must consider all other outside network access, including dial-in lines and SLIP/PPP connections.

Tri-Homed Hosts

A tri-homed host combines elements of a screening router and a screened host, thereby overcoming the limitations of each. Security is centered on the screening routers by using interfaces for the Internet, the intranet, and the subnets that contain the bastion hosts and application servers. An example of a tri-homed host is shown in the following figure.

Figure 1-5 *Tri-Homed Host*



1.3.2 Firewall Technologies

The technologies that create a firewall service include packet filtering, NAT, circuit-level gateways, application proxies, and VPNs. The security and efficiency of these technologies and services vary, depending on their efficiency and sophistication.

- "Packet Filtering" on page 21
- "Network Address Translation" on page 22
- "Circuit-Level Gateways" on page 22
- "Application Proxies" on page 22
- "Virtual Private Networks" on page 22

Packet Filtering

Packet filtering is the basic method for protecting the intranet border. Packet filters work at the Network layer of the OSI model. The limitation of packet filtering is that it cannot distinguish usernames.

Packet filters filter data based on service type, port number, interface number, source address, and destination address, among other criteria. For example, a packet filter can permit or deny service advertisements on an interface. You can use incoming and outgoing filters to dictate what information passes into or out of your intranet.

Network Address Translation

Network Address Translation (NAT) maps private IP addresses to public IP addresses. NAT can perform this mapping both dynamically and statically. An alternative to NAT is a circuit-level gateway.

Circuit-Level Gateways

A circuit-level gateway works at the Session layer in the OSI model, which means that even more information is required before packets are allowed or denied. Access is determined based on address, DNS domain name, or the NDS or eDirectory username. Special client software must be installed on the workstation. Circuit-level gateways can bridge different network protocols, such as, IPX to IP.

Your username is checked and granted access before the connection to the router is established. Compare this with NAT, where only the private source IP address is used to gain access. NAT performance is greater than circuit-level gateway performance but circuit-level gateways are more secure.

Application Proxies

In a circuit-level gateway, after a virtual pipe is established between the client and host, any application can be used across the connection. The reason is that circuit-level gateways cannot determine the application-level contents of packets being sent between the client and host during a transmission. An application-level proxy works as a proxy server to intercept information running across a gateway, preventing direct communication between the client and the host.

An application proxy is specific to an application, for example, an FTP proxy or SMTP proxy. An application proxy accepts only packets that are generated by protocols that the proxy can copy, forward, and filter.

Virtual Private Networks

Virtual Private Networks (VPNs) allow two hosts to exchange data using a secure channel. The data stream is encrypted for security. A VPN can be configured as a connection between two endpoints or between many endpoints. You can connect two offices over an Internet connection, or connect several offices to create a secure private network. Remote VPN clients are also supported.

1.3.3 Novell BorderManager Firewall Solutions

This section describes the firewall services provided by Novell BorderManager, and explains how you can effectively use Novell BorderManager as part of a comprehensive and versatile solution to your security policy.

Novell BorderManager firewall services provide increased security through three levels of firewall protection, including packet filtering (Level I firewall), circuit-level gateways (Level II firewall), and application proxy services (Level III firewall). In addition, the Network Address Translation (NAT) services can allow unregistered intranet IP addresses to connect to the Internet and, at the same time, conceal these addresses from outsiders. Also, the Virtual Private Network (VPN) services deliver secure, encrypted connections over the Internet, eliminating expensive leased lines.

A firewall can consist of several components. Novell BorderManager provides a comprehensive firewall solution that includes the following services:

- Packet filtering
- Network Address Translation (NAT)
- Proxy Services
- Access control
- CyberPatrol*
- SOCKS 5 for authentication and SSL for encryption services
- Virtual Private Network (VPN)
- Alerts for specific server conditions

IMPORTANT: To use Novell BorderManager to secure your network border, you must specify a public IP address. Public IP addresses specify server interfaces to a public network, typically the Internet. Public network interfaces are not secure. Private IP addresses specify server interfaces to a private network, or intranet.

Packet Filtering

Packet filters provide network-level security at the router by permitting or denying packets based on a predefined set of rules. Novell BorderManager supports Routing Information Protocol (RIP) filters and packet forwarding filters to control the service and route information for the common protocol suites, including IPX and TCP/IP.

A packet filtering router, for example, can filter IP packets based on the source or destination IP address and the TCP/UDP port number, and can filter based on the source or destination interfaces. Filters can also block connections to or from specific hosts or networks and to specific ports.

Specific services and protocols can also be filtered. For example, X Window System*, RPC, and rlogin services should be blocked because they can create an open threat to corporate security. Refer to Chapter 2, "Packet Filtering Overview and Planning," on page 29 for more information on packet filters.

Network Address Translation

NAT does not require special client software and can be used by hosts on any platform that use the gateway as a route to the Internet. NAT enables any IP host on your local network to access the Internet without you assigning a globally unique IP addresses to each system.

Novell BorderManager provides both dynamic and static IP network address translation. For static IP address translation, tables are defined with sets of public IP addresses. These tables are then used to map the source addresses of packets being sent through the firewall to their public addresses.

NAT also acts as a filter, allowing only certain outbound and inbound connections. The type of filtering that occurs is determined by whether NAT is configured to operate in dynamic or static mode.

For detailed information on NAT, refer to Chapter 4, "NAT Overview and Planning," on page 53.

Proxy Services

Proxy Services provides application-level security by providing application proxies that forward and filter connections for such services as HTTP, Gopher, FTP, SMTP, RealAudio*, and DNS. In general, Proxy Services only allows services for which there are proxies. For example, if a gateway has a proxy for FTP, then only FTP is allowed into the protected subnet; all other services are blocked.

TTP proxy improves performance by locally caching frequently requested Internet information and optimizing WAN bandwidth use. The client (browser) makes a request directly to a proxy, which locates the object in its cache and returns the object to the client. If the object is not in the cache, the proxy retrieves it from the origin Web server on the Internet, stores it in the cache, and returns the object to the client. Benefits include reduced Internet traffic and reduced request load on the object source, which in turn reduce delays in returning information to the client.

Proxy Services also allows protocol filtering. You can set up the firewall to filter FTP connections and deny use of the FTP put command. This can be useful if you do not want users writing to an anonymous FTP server.

When using the proxy server (or gateway), you can hide the names and addresses of internal systems—the gateway is the only hostname known outside the system. Also, traffic can be logged before it reaches the internal hosts. Proxy Services improves security by hiding private network domain names and addresses and sending all requests through a single gateway.

For detailed information on Proxy Services, refer to Chapter 5, “Proxy Services Overview and Planning,” on page 63.

Access Control

Access control controls Internet and intranet access at the Application layer by allowing or denying access requests made through proxy servers and VPNs. By controlling access at the Application layer, you can attain a higher level of security than you can with packet filtering, which controls access only at the Network layer. Access control can also use usernames instead of source or destination IP addresses.

Access control involves establishing a set of rules. Each time an access request is made, the Novell BorderManager server searches for the rules that apply to the request. If no rule is found, the request is denied (the default). You can create access control rules at the Country, Organization, Organizational Unit, and Server object levels. Rules can be based on criteria such as users, groups, IP addresses, or services. With access rules, you can control access to network and Web services, proxy services, VPNs, and URLs.

In general, firewall security should provide the following levels of access control:

- Host control: Determine which hosts can be accessed.
- Application-level control: For example, allow access to the Web but prevent access to newsgroups.
- Content control: Determine which network files and information can be accessed.

For detailed information on access control, refer to Chapter 3, “Access Control Overview and Planning,” on page 41.

Virtual Private Networks

A VPN is used to transfer sensitive company information across an untrusted network, such as the Internet, in a secure fashion by encapsulating and encrypting the data. For site-to-site VPN, only the VPN members must be running the VPN software; for client-to-site VPN, the VPN client must also be running the VPN software.

Client-to-site VPNs can use two types of secure connections:

- Direct dial-in connections
- Internet Service Provider (ISP) connections through the Internet

Site-to-site VPNs can use the following types of secure connections:

- Between two departments within the same company using the company's private network, or intranet
- Between two or more sites within the same company using the Internet
- Between two or more different companies using the Internet

Both intranet and Internet site-to-site VPNs can be deployed in one of two ways:

- With the VPN member on the border between your private network and the public network
- With the VPN member behind a high-end router that is on the border between your private network and the public network

1.4 Improving Network Performance

This section describes how you can improve network performance using Novell BorderManager Proxy Services and caching.

- "Improving Performance by Using Proxy Services" on page 25
- "Using Different Types of Caching to Improve Performance" on page 26

1.4.1 Improving Performance by Using Proxy Services

Next to security, performance is of major concern. When you connect to the Internet, performance problems can occur for many reasons, including the following:

- Connections to the Internet are slower than connections within the intranet.
- Internet users can access enormous amounts of information.
- The same information is being accessed multiple times, and replicated information is clogging the networks.
- Large graphics, video, and audio files are typical for Web sites, take up more space, and take longer to download than plain text.

Novell BorderManager Proxy Services can greatly improve network performance by locally caching frequently requested Internet information. In general, Proxy Services stores copies of frequently requested Web information closer to the user, thereby reducing the number of times the same information is accessed over an Internet connection, the download time, and the load on the remote server.

Novell BorderManager provides advanced caching technologies to address the performance issues. These are discussed in the next section. For more detailed information on Proxy Services, refer to Chapter 5, "Proxy Services Overview and Planning," on page 63.

1.4.2 Using Different Types of Caching to Improve Performance

There are three primary ways to use caching to improve performance:

- "Web Client Acceleration (Standard Proxy Cache)" on page 26
- "Web Server Acceleration (Reverse Proxy Cache or HTTP Acceleration)" on page 26
- "Network Acceleration (Hierarchical Caching)" on page 27

Web Client Acceleration (Standard Proxy Cache)

In Web client acceleration, the proxy server is located between the clients and the Internet or intranet. The proxy server intercepts requests from clients for Web pages and supplies the requested pages, if cached, to the client at LAN speed. This eliminates the delay that occurs when the origin Web site is accessed using a slower link, and it also minimizes the traffic between the corporate network and the Internet or intranet.

The proxy server works with the NDS or eDirectory access control policies to make sure that the client browser has authorization to access the data.

Use Web client acceleration to improve performance for sites that have the following:

- Multiple clients
- Remote locations that route through a central or main site to access the Internet
- A need for access control at the user level
- A need for globally and centrally maintained multiple firewall services

Web Server Acceleration (Reverse Proxy Cache or HTTP Acceleration)

With Web server, or HTTP, acceleration, the proxy server acts as a front end to one or more publishing Web servers and caches all information that belongs to the Web server. When a client requests information from a Web server, the request is redirected to the proxy server (the user enters a URL to the accelerator server rather than to the origin host). The proxy server supplies the cached pages to the client at high speed. This method accelerates access and takes the request load off the Web servers, enabling them to respond to more users.

The reverse proxy cache server is an automatic firewall to the publishing servers. The reverse proxy cache server caches all static information (typically up to 90 percent of a Web site), freeing up the publishing server to open connections and deliver the dynamic data. It also protects the IP addresses of the origin servers, thereby increasing security.

Use reverse proxy cache for sites that have the following:

- Frequently used Web servers
- A need for centralized access control on all intranet Web servers
- Intranet publishing servers that store public and private data
- A mix of Internet and intranet Web server platforms

Network Acceleration (Hierarchical Caching)

With network acceleration, or hierarchical caching, multiple proxy servers are configured in a hierarchical topology. The proxy servers are connected in a parent, child, or peer relationship. When a miss occurs, the proxy contacts the other servers in the hierarchy to find the requested cached information. The nearest proxy cache or the proxy cache with the highest assigned priority that has the requested information forwards it to the requesting proxy server, which in turn forwards it to the requesting client.

Hierarchical caching reduces the WAN traffic load and increases valuable bandwidth. In addition, because the requested information is sent from the nearest proxy server, network delays are minimized. This reduces user wait time and increases user productivity.

Use hierarchical caching for sites that have the following:

- Slow Internet links
- Multiple LAN segments that can be used to store data closer to users
- Congestion or delay at LAN points in the WANs

1.5 Managing Novell BorderManager Services

Novell BorderManager enables you to manage all Novell BorderManager servers from a centralized location using familiar NetWare tools. Novell BorderManager provides this essential functionality by closely integrating the Novell BorderManager services with the NDS or eDirectory database. Therefore, you can enforce and monitor access consistently from a central location.

- "Management through eDirectory" on page 27
- "Event Logging and Auditing" on page 27
- "Alerts" on page 28

1.5.1 Management through eDirectory

Management through eDirectory has the following features:

- A single point of administration
- Integration for centralized access control management
- Proxy authentication

eDirectory can also be used to enforce periodic changes to passwords, specify a minimum password length, and enforce alphanumeric combinations. The eDirectory directory provides access control to restrict objects in the database.

NOTE: When configuring eDirectory, do not replicate the directory to a system that is not physically secure. An eDirectory replica on a machine that can be physically disconnected from the network is subject to offline attack.

1.5.2 Event Logging and Auditing

A major task in enforcing network security is monitoring event logs on regular weekly intervals. These logs are used to check for anomalies in server traffic, such as port scans, spoofed Routing

Information Protocol (RIP) packets, Domain Name System (DNS) requests, ICMP redirects, or any inconsistent routing activity. A history log file can be an excellent tool for identifying irregular patterns.

Audit your network for devices and validate their use. Check for unauthorized modems and network traffic analyzers.

1.5.3 Alerts

You can configure Novell BorderManager Alert to notify you by e-mail when certain conditions or events occur on your Novell BorderManager server. Performance-related conditions or events include memory shortages, disk space shortages, and down servers. Security-related conditions or events include packet flooding, abnormal packet sizes, and the unloading or loading of security-sensitive modules on the server. For more information, see *Novell BorderManager 3.9 Administration Guide*.

1.6 Where to Go from Here

How you approach the rest of the information in this documentation depends on how much you already know about NetWare and Novell BorderManager, and whether you need more detailed information to understand the Novell BorderManager services. Each service-specific chapter provides details about how the service operates and explains when to use the service. The following table indicates what you should read, depending on the information you require.

For information on	Read
Packet filtering	Chapter 2, “Packet Filtering Overview and Planning,” on page 29
Access control	Chapter 3, “Access Control Overview and Planning,” on page 41
NAT	Chapter 4, “NAT Overview and Planning,” on page 53
Proxy Services	Chapter 5, “Proxy Services Overview and Planning,” on page 63
Example scenarios	Chapter 6, “Novell BorderManager Planning Scenarios,” on page 93

Packet Filtering Overview and Planning

2

This section contains overview and planning information for Novell® BorderManager® 3.9 packet filtering. It contains the following sections:

- Section 2.1, “Overview of Packet Filtering,” on page 29
- Section 2.2, “How Packet Filtering Works,” on page 30
- Section 2.3, “Monitoring Packet Filtering,” on page 36

2.1 Overview of Packet Filtering

The Internet is increasingly becoming an accepted medium for conducting business transactions. Your company, like many others, needs to connect its private data network (or intranet) to the public network (or Internet) to interact with customers, suppliers, and business partners. The World Wide Web provides expanded facilities for electronic commerce, as well as readily available remote access for telecommuters and other mobile workers. Intracompany Web sites are used to provide information on everything from employee benefits to technical support. The Internet can provide cost savings in communications; however, it can also be a source of new and increased security risks.

To reduce the security risks inherent in connecting to the Internet, or in providing remote access to your internal networks, appropriate network security policies must be defined as part of your routine business strategy. Novell BorderManager provides enhanced packet filtering capabilities that can be used to build firewalls that can enforce your access policies.

A firewall is a network component that controls the traffic flowing between internal (private) networks and external (public) networks, such as the Internet. Firewalls can also be used to separate your internal data networks (intranets) to protect valuable company data—research and development, corporate financial data, personnel files, and other sensitive information.

Novell BorderManager protects your confidential information from internal and external intruders with its advanced security services. Novell BorderManager packet filtering provides a basic level of network security by controlling both Internet and intranet access at the network level.

This section describes how packet filters can be used to ensure that all traffic is routed securely through your Novell BorderManager server. It contains the following subsections:

- Section 2.1.1, “Packet Filtering Security Options,” on page 29
- Section 2.1.2, “Other BorderManager Security Services,” on page 30

2.1.1 Packet Filtering Security Options

Typically, users in your organization need e-mail access, Internet access, and remote access. These services can represent security threats to your network because they involve the transmission and reception of packets across the border between your private network (intranet) and the public network (Internet) or other external sources.

Packet filtering provides a number of security options for Novell BorderManager servers, including the following:

- Packet filtering systems guard against security attacks such as address spoofing, in which an unauthorized host sends false addresses to gain network access.
- Packet filtering prevents unauthorized network access without interfering with authorized access.
- Strategically placed packet filters can protect your entire network from a wide variety of denial-of-service attacks.

As the first line of defense, Novell BorderManager packet filtering takes a fairly simple approach to network security: it rejects all packets except those that you explicitly instruct it to allow to pass. Your Novell BorderManager server's packet filters should be set up to reject unwanted packets and pass all other packets to higher-level, more secure measures, such as a circuit-level gateway or an application gateway.

Two significant benefits can be derived from filtering packets. In addition to protecting your private network from unwanted intruders, you can achieve a significant reduction in traffic. If your Novell BorderManager server is set up to pass packets to a circuit-level gateway or an application gateway, the fewer packets these components must process, the better performance you will have.

However, packet filters can be difficult to manage. As your set of packet filter rules grows more complex, it becomes easier to generate conflicting rules or mistakenly allow unwanted data in to or out of your intranet. To avoid this problem, your organization must have a security policy that clearly defines the traffic that is to be allowed through the Novell BorderManager server.

2.1.2 Other BorderManager Security Services

Although packet filters are a prerequisite for securing your corporate network, or intranet, from outsiders, you should be aware that packet filtering alone cannot provide adequate protection. Packet filtering is just one security mechanism that can be used to control data transfer to and from the public network, or Internet.

In a typical firewall architecture, the interface that is connected to the external (public) network forces all inbound traffic to pass through the Novell BorderManager server. The interface that is connected to the internal (private) network forces all outbound traffic to pass through the Novell BorderManager server. The packet filter rules set up on the Novell BorderManager server control what type of packets are allowed to pass.

Novell BorderManager firewall services provide increased security through three levels of firewall protection, including packet filtering (Level I firewall), circuit-level gateways (Level II firewall), and application proxy services (Level III firewall).

Novell BorderManager servers can implement security policies by application or by user. The Novell BorderManager server controls the delivery of network-based services both to and from the internal network. For example, only certain users are allowed to communicate with the Internet, or only certain applications are permitted to establish connections between internal and external hosts.

2.2 How Packet Filtering Works

Novell BorderManager implements packet filtering in the following ways:

- Section 2.2.1, "Static Packet Filtering," on page 31

- Section 2.2.2, “Advanced Features,” on page 31

2.2.1 Static Packet Filtering

Static packet filtering systems, such as the one used in earlier releases of Novell BorderManager, examine each packet that crosses the border between your private network (intranet) and the public network (Internet). Static packet filters parse the header field of each packet to identify a set of characteristics:

- Protocol ID (for example, TCP, UDP, ICMP)
- Source IP address and port number
- Destination IP address and port number
- Router interface for the incoming or outgoing packet

These characteristics determine whether the packet should be forwarded in accordance with fixed sets of outbound and inbound rules. Each filter set handles traffic coming to the firewall over a specific interface. Port numbers are used to filter traffic. Connections for various services, such as e-mail, FTP, Telnet, and so on, are denied by filtering the packets attempting to use that service or port number.

Firewalls based on static packet filtering are Network-layer devices that cannot process higher-layer information. They cannot check for application requests, nor can they keep track of application state information. A static packet filtering firewall cannot determine, simply by examining the header of an incoming packet, whether the packet is the first packet from an external client to an internal server or a response from an external server to an internal client. The level of protection provided by this type of firewall is limited.

2.2.2 Advanced Features

Novell BorderManager has three advanced IP packet filtering features to enhance firewall security:

- “TCP ACK Bit Filtering” on page 31
- “Dynamic Packet Filtering” on page 32
- “Fragmented Packet Filtering” on page 35
- “Defense Against Other Attacks” on page 35

TCP ACK Bit Filtering

TCP is a connection-oriented and reliable transport protocol. A connection is always initiated using the well-known three-way handshake, as follows:

Packet 1: Client –> ServerFlag: SYN

The client wants to initiate a connection or synchronization.

Packet 2: Server –> ClientFlags: SYN, ACK

ACK: The client’s connection request [SYN] is being acknowledged.

Packet 3: Client –> ServerFlag: ACK

ACK: The server’s connection request [SYN] is being acknowledged.

When TCP ACK bit filtering is enabled, only response packets (packets with the TCP ACK bit set) are allowed through. This effectively blocks all connection attempts from being initiated through this filter rule. TCP ACK bit filtering is often applied to all TCP inbound filters to prevent external hosts from initiating TCP connections to internal hosts; however, it should not be used in the following circumstances:

- An internal service is provided to an external host (because the first TCP connection packet does not have the ACK bit set).
- TCP applications, such as FTP, require incoming connections.
- Packets, such as UDP packets, do not have an ACK bit.

IMPORTANT: Stateful packet filtering is a superset of ACK bit filtering; however, they cannot be configured on the same filter.

Dynamic Packet Filtering

Dynamic, or stateful, packet filtering, is designed to overcome the limitations of static packet filtering. Dynamic packet filtering tracks the outgoing packets it has allowed to pass and allows only the corresponding response packets to return. When the first packet is transmitted to the public network (Internet), a reverse filter is dynamically created to allow the response packet to return. To be counted as a response, the incoming packet must be from the host and port to which the outbound packet was sent.

Stateful packet filtering supports both connection and connectionless protocols (TCP, UDP, ICMP, and so on). Dynamic packet filtering monitors each connection and creates a temporary (time-limited) inbound filter exception for the connection. This allows you to block incoming traffic originating from a particular port number and address while still allowing return traffic from that same port number and address. The reverse filter is created by extracting the following packet information:

- Source IP address
- Source interface
- Source port
- Destination IP address
- Destination interface
- Destination port
- Protocol type

This information is stored in a table, which is compared against the reply. If an incoming message is not a reply to the original request, then it is dropped. This dynamically created filter set is used to determine the subsequent packet transfers until the connection is closed.

For connection-oriented protocols, such as TCP, the incoming (reverse) filter is created only when the first outgoing packet is detected. TCP ACK bit filtering is automatically enabled on the reverse filter to prevent any connection attempts by intruders from being initiated through that filter.

For ICMP, only the reply ICMP messages are allowed. All ICMP requests and ICMP redirect messages that attempt to return through a reverse filter are dropped. All other ICMP error reply messages are handled internally by stateful packet filtering.

Without stateful packet filtering, you would need to create two ICMP filters: one to allow an ICMP error message to go out and the other to allow it to return. With stateful packet filtering, you no longer need to create these ICMP filters. Without stateful packet filtering, you would need four sets of outbound and inbound filters (a total of at least eight separate filters) to establish FTP service.

The following four tables list the settings for each outbound and inbound filter:

Table 2-1 *Inbound and outbound filter settings*

Control Channel	Outbound	Inbound
Source Interface	Private	Public
Destination Interface	Public	Private
Packet Type	TCP	TCP
Source Port	1024-65535	21
Destination Port	21	1024-65535
Source Address	Any	Any
Destination Address	Any	Any
ACK Bit Filtering	Disabled	Enabled
Stateful Filtering	Disabled	Disabled

Table 2-2 *Inbound and outbound filter settings*

PORT Command Data Channel	Outbound	Inbound
Source Interface	Private	Public
Destination Interface	Public	Private
Packet Type	TCP	TCP
Source Port	1024-65535	20
Destination Port	20	1024-65535
Source Address	Any	Any
Destination Address	Any	Any
ACK Bit Filtering	Enabled	Disabled
Stateful Filtering	Disabled	Disabled

Table 2-3 *Inbound and outbound filter settings*

PASV Command Data Channel	Outbound	Inbound
Source Interface	Private	Public
Destination Interface	Public	Private

Packet Type	TCP	TCP
Source Port	1024-65535	1024-65535
Destination Port	1024-65535	1024-65535
Source Address	Any	Any
Destination Address	Any	Any
ACK Bit Filtering	Disabled	Enabled
Stateful Filtering	Disabled	Disabled

Table 2-4 *Inbound and outbound filter settings*

ICMP Error Messages Channel	Outbound	Inbound
Source Interface	Private	Public
Destination Interface	Public	Private
Packet Type	ICMP	ICMP
Source Port	N/A	N/A
Destination Port	N/A	N/A
Source Address	Any	Any
Destination Address	Any	Any
ACK Bit Filtering	N/A	N/A
Stateful Filtering	Disabled	Disabled

With stateful packet filtering, you need only one outbound filter to establish FTP service, as shown in the following table:

Table 2-5 *Outbound filters*

FTP Filter	Outbound
Source Interface	Private
Destination Interface	Public
Packet Type	TCP
Source Port	1024-65535
Destination Port	21
Source Address	Any
Destination Address	Any
ACK Bit Filtering	Disabled
Stateful Filtering	Enabled

FTP PORT Command Support

Stateful packet filtering supports active FTP sessions by monitoring the `PORT` command. The client application sends this command over the FTP command channel (port 21) to signal the server on which port the client is listening for the server to open the data channel. Stateful packet filtering creates a temporary (time-limited) exception rule to allow the server to open the data channel (usually from port 20) to the client.

FTP PASV Command Support

The difference between the passive FTP and the active FTP modes is that in the passive mode the client is allowed to open both the FTP command (port 21) and FTP data channels to the server. The passive FTP client always initiates the connection to the server's command and data channels, and the server only needs to signal the client on which port to connect back to the server's data port, which is normally above 1023.

Stateful packet filtering monitors the `PASV` command instead of the `PORT` command, and it creates a temporary (time-limited) exception filter to allow the client to communicate to the server on a designated high port (above 1023). With stateful packet filtering, all data and ICMP channels and the reverse path for the control channel are created (and eliminated) dynamically.

NOTE: For FTP service, you can choose which mode of FTP to allow: PORT, PASV, or ENABLED (which allows both PORT and PASV modes). Some companies do not allow internal users to use the active FTP (PORT) service.

Fragmented Packet Filtering

When an IP datagram is fragmented, only the first packet has the complete header and transport information. All subsequent packets do not have the transport information, such as the port number. Formerly, it was safe to allow these fragments to pass through the firewall unchecked, as long as the first packet was dropped. The target host would eventually drop all subsequent packets, and reassembly could not be completed without the first packet.

This is no longer true today. Fragmented packets can be used to launch denial-of-service attacks by continuously flooding the network with fragment packets to consume network bandwidth and resources on the target host. Tiny fragments can also be used to bypass the firewall with overlapped fragments or by manipulating the fragment flags.

For example, if a firewall does not impose its security on packet fragments, then all packets with the fragment bit set can pass through unchecked. A nonfragmented packet with the more fragment bit set can be used to perform a port scan on the target host. To prevent such attacks, all fragments must be checked based on source and destination addresses, inbound and outbound interfaces, and a combination of TCP/IP and fragment flags.

To protect against fragment attacks with manipulation of the fragment flags, Novell BorderManager has added automatic fragmented packet filtering. Fragmented packet filtering discards the first packet, which carries the complete header and transport information, and all subsequent fragmented packets if they have the same source and destination IP addresses and interfaces.

Defense Against Other Attacks

Defenses have been built into the Novell BorderManager software for various attack programs that currently exist on the Internet. The following additional commands can also be enabled (or disabled) on the server console through `SET` commands under the COMMUNICATION category, or they can be included in your `autoexec.ncf` startup file. By default, they are set to ON.

- `Set Filter Subnet Broadcast Packets=ON/OFF`

  When set to ON, all packets with a destination IP broadcast address are dropped.

  The address 255 is used for broadcasts. A broadcast is a message that is sent to every system on the network. It is often easier to send a single broadcast message than it is to send individual datagrams to each host. Subnet broadcast messages are sent using an address that consists of the network address with 255 in the subnet number portion of the address. For example, if you are on network 140.120.8, the IP address 140.120.8.255 would be used for subnet broadcasts.
- `Set Filter Local Loopback Packets=ON/OFF`

  When set to ON, all local loopback packets are dropped.

  This command has been added to support local applications that use the local loopback address to run locally on the server, such as the interfaces on the NetWare 5 software, NDPS™ Broker, and the ConsoleOne® software.
- `Set Filter Packets with IP Header Options=ON/OFF`

  When set to ON, all packets with an IP header option enabled are dropped.

  This command has been added to prevent the forwarding of packets to unauthorized hosts through the source routing option, which lists the IP addresses to which the packet must be forwarded.

2.3 Monitoring Packet Filtering

To view all the filters that you have created, you can save the filter information to a text file. To create this file, load FILTCFG and select *Save Filters to a Text File* from the *Filter Configuration Available Options* menu. You can save the file to any name you prefer, such as MYFILTER.

You can also monitor the operation of the filters you have created to ensure that they are actually filtering the types of packets that you intended for them to filter. For more information on packet filter logging, see *Novell BorderManager 3.9 Administration Guide*.

- Section 2.3.1, “Packet Filtering Security,” on page 36
- Section 2.3.2, “Filter Action Options,” on page 38

2.3.1 Packet Filtering Security

Because packet filtering does not inspect the packet’s Application-layer data, this solution is the least secure but most efficient of the firewall methods. If the checks are passed successfully, the packet is allowed to be routed through the firewall. However, because this approach requires less processing than the other methods, it is the fastest solution.

Packet filtering has the following advantages:

- Client computers require no specific configuration.
- Packet filters are faster because they perform less processing.
- A single filter rule can deny traffic between internal and external sources.
- Packet filters can accept or reject packets according to well-known protocol port numbers (such as TCP port numbers).

Packet filtering has the following limitations:

- Packet filters have no alert capability.
- Packet filter rules can be difficult to configure.

Two basic security policy philosophies can be applied in packet filtering:

- Deny everything that is not permitted.
- Permit everything that is not denied.

The default packet filtering mode (secure mode), which is normally selected during Novell BorderManager installation, takes the first approach—deny everything. This is the better choice when you initially set up your Novell BorderManager server because you are more likely to make mistakes that could compromise security when you first install and configure the server.

When Novell BorderManager is installed, a set of default filters prevents access to the Internet without the services of an application proxy or a gateway, as listed in the following table:

Table 2-6 *Default packet filtering mode*

Filter Type	Protocol	Setting
IPX™ Filters		
	Outgoing (to Public interface)	
	SAP	Deny service name * and service type FFFFh (All)
	RIP	Deny Network 00000000h and mask 00000000h
	Packet Forwarding	Deny All packets
	Incoming (from Public Interface)	
	SAP	Deny service name * and service type FFFFh (All)
	RIP	Deny Network 00000000h and mask 00000000h
	Packet Forwarding	Deny All packets
IP Filters		
	Outgoing (to Public interface)	
	RIP	Do not advertise all routes
	EGP	Do not advertise all routes
	Packet Forwarding	Deny all packets
	Incoming (from Public Interface)	
	RIP	Do not advertise all routes
	EGP	Do not advertise all routes
	Packet Forwarding	Deny all packets
OSPF Filter	OSPF	Deny all routes

Filter Type	Protocol	Setting
Exception Filters		
	Outgoing (to Public interface)	
	Packet Forwarding	Allow All packets with destination IP address as the public IP address
	Incoming (from Public Interface)	
	Packet Forwarding	Allow All packets coming from the public interface with source IP address as the public IP address and destined to the following ports and protocols:
		TCP port 443: SSL Authentication
		TCP ports 1024 to 65535: Dynamic TCP
		UDP ports 1024 to 65535: Dynamic UDP
		TCP port 213: VPN Master/Slave (IPX)
		TCP port 353: VPN Authentication Gateway
		UDP port 353: VPN Keep-Alive
		SKIP (Simple Key Management for Internet Protocol) protocol 57—for VPN
		TCP port 80—World Wide Web (WWW)—HTTP

TIP: The Novell BorderManager default filter settings block most traffic into and out of the server until you can configure filters that allow specific types of packets to pass. For this reason, we recommend that you set up and configure packet filters after normal business hours to avoid interruption of network traffic.

Packets must be expressly permitted, and they must not be expressly denied. However, you can make exceptions to either of these conditions through iManager. After the packet data is obtained, the filter applies lists of *rules*: first the exception list, then the filter list. These lists determine what packets can flow to and from the network.

2.3.2 Filter Action Options

Filtering rules in the exception lists and filter lists are applied using one of two filter action options, Deny or Permit.

Deny

If the filter action option is set to *Deny Packets in Filter List*, the filter list contains the list of packets to deny and the exception list contains the list of packets to permit. Exception filters always take priority over deny filters. If a packet type is not listed in the exception filter list, it is checked against the deny filter list. If the packet type is not listed in either list, it is allowed.

Permit

If the filter action option is set to *Permit Packets in Filter List*, the filter list contains the list of packets to permit and the exception list contains the list of packets to deny. Exception filters always take priority over permit filters. If a packet type is not listed in the exception filter list, it is checked against the permit filter list. If the packet type is not listed in either list, it is denied.

These two filter action options can be summarized as shown in the following table.

Table 2-7 *Filter Action Options*

Filter Action	Description
Deny	All packets specified in the exception list are permitted. All packets specified in the filter list are denied. If the Deny mode is enabled and no filters are specified, the router permits all packets to pass.
Permit	All packets specified in the exception list are denied. All packets specified in the filter list are permitted. If the Permit mode is enabled and no filters are specified, the router does not permit any packets to pass.

Access Control Overview and Planning

3

This section contains overview and planning information for Novell® BorderManager® 3.9 access control. It contains the following sections:

- Section 3.1, "Overview of Access Control," on page 41
- Section 3.2, "Implementing Access Control," on page 47
- Section 3.3, "Monitoring Access Control," on page 51

3.1 Overview of Access Control

Access control is a key part of Novell BorderManager security. This section provides information about using access control, and describes what access rules are, what an access control list is, and how access control works on a Novell BorderManager server. It contains the following subsections:

- Section 3.1.1, "Using Access Control with BorderManager," on page 41
- Section 3.1.2, "Implementing Access Control," on page 42
- Section 3.1.3, "Access Rules," on page 42
- Section 3.1.4, "Using Wildcards in Access Rules," on page 44
- Section 3.1.5, "Access Control Lists," on page 45

3.1.1 Using Access Control with BorderManager

Access control is the process by which an administrator can regulate and monitor user access to intranet or Internet services. Access control supplies a much broader range of network security options than packet filtering alone. Like packet filtering, you can use access control to provide network-level security (Level I firewall), but you can also use it to provide circuit-level security (Level II firewall) and application-level security (Level III firewall).

In addition, you can use access control to implement an overall security policy that is customized in far more detail to meet the needs of your site. You can define access by user, user group, time of day, application, and so on.

You use access control to specify which requests made through the Proxy Services, and Virtual Private Networks (VPN) should be allowed and which requests should be denied. For example, you might want to control the use of a particular proxy service, or you might want to limit who can connect to the Internet and specifically when they can make the connection.

By configuring access control to use SurfControl, you can also create access rules that use SurfControl's URL categories. This enables you to implement category-based content filtering to control user access to entire categories of URLs that might contain objectionable Web content. For more information, see the SurfControl Web site at (www.surfcontrol.com).

Novell BorderManager access control software works by allowing or denying access requests made through the Proxy Services, and VPN. For information about configuring these services, see *Novell BorderManager 3.9 Administration Guide*.

3.1.2 Implementing Access Control

Novell BorderManager access control consists of two elements:

- **Access rule:** Specific rules that define which sources can access which destinations at what times.
- **Access control lists:** Ordered sets of access rules that control intranet and Internet access through a Novell BorderManager server.

3.1.3 Access Rules

Access rules are the primary elements of access control. They apply to clients requesting access through the Proxy Services, or a VPN. Access rules control traffic through a Novell BorderManager server, usually between your company's intranet and the Internet.

By creating Allow or Deny rules, you can control access to the following:

- Many network and Web services
- Novell BorderManager Proxy Services
- VPNs
- URLs

You can configure access rules at the following NDS® or Novell eDirectory™ object levels:

- Country (C)
- Organization (O)
- Organizational Unit (OU)
- Server

You can apply access rules across a wide range of users and resources by specifying rules for Organizations and Organizational Units. You can also create rules that apply specifically to individual users, IP addresses, and so on.

We recommend that you configure rules affecting all eDirectory users in a container high enough in the eDirectory tree to include all Server objects representing servers running Novell BorderManager. This ensures that the same rules are applied to all eDirectory users, irrespective of which Novell BorderManager server they use to access the Internet.

Access Rule Structure

When you create an access rule, you can specify the following elements:

- "Action" on page 43
- "Access Type" on page 43
- "Source" on page 43
- "Destination" on page 44
- "Time Restriction" on page 44
- "Rule Hit Logging" on page 44

Action

The Action parameter specifies how the access rule functions, as follows:

- Allow—An Allow rule specifies that particular access requests are to be allowed based on the source, access type, destination, or time restrictions specified.
- Deny—A Deny rule specifies that particular access requests are to be denied based on the source, access type, destination, or time restrictions specified.

Access Type

The Access Type selection is the most important control element in an access rule. You can specify one of the following:

- **Port:** An access request made to a specific port based on well-known TCP/IP port numbers. For port access requests, you can specify the type of port, the port numbers, and the transport protocol (TCP, UDP, or TCP & UDP).
- **URL:** An access request made to a specific URL based on a Web site or Web page.
- **Application Proxy:** An access request made to a proxy server based on the proxies implemented in Novell BorderManager (HTTP, FTP, SMTP, NNTP, and Generic TCP/UDP). The control of application proxies is based on well-known TCP/IP port numbers, but it is more specific to the protocol used by the Novell BorderManager Proxy Services component.

  For proxy access rules, you can specify the type of proxy, the port number, the direction of the requests (Send, Receive, or Send & Receive), and details about file types you want to restrict.
- **VPN Client:** A request made for access to a VPN server.

Source

The Source parameters represent users who want to access the company's intranet or the Internet through a Novell BorderManager server. Each transaction on a network has a source and a destination. Depending on the Access Type selection you specified previously, the access rule configuration utility allows you to configure access rules that apply to different sources, as follows:

- <Any> All users
- One or more eDirectory users
- One or more eDirectory user groups

  Specific eDirectory usernames, user groups, or containers are used to match an eDirectory name to an access request. Novell BorderManager stores the eDirectory names of users, groups, and containers as fully qualified typeless names. The access control configuration utility displays eDirectory objects for you to choose. You cannot manually type eDirectory names into access rules.
- One or more DNS hostnames

  Specific hostnames are used to match the DNS name in a user's access request.
- One or more e-mail usernames or e-mail domain names

  Specific e-mail usernames or e-mail domain names are used to match the user's e-mail name and domain name in an access request.
- An IP address or a range of IP addresses
- One or more IP subnet addresses, including each subnet address's subnet mask (255.255.252.0, for example)

Specific IP addresses, a range of IP addresses, or IP subnet addresses are used to match the IP address of the user's browser in an access request.

Destination

The Destination parameters represent the intranet or Internet host the user wants to access through the Novell BorderManager server. Depending on the Access Type selection you previously specified, the access rule configuration utility allows you to configure access rules that apply to different destinations, as follows:

- <Any> All users
- One or more DNS hostnames

  Specific hostnames are used to match the DNS name of the Web site the user wants to access.
- An IP address or a range of IP addresses
- One or more IP subnet addresses, including each subnet address's subnet mask (255.255.252.0, for example)

  Specific IP addresses, a range of IP addresses, or IP subnet addresses are used to match the IP address of the Web site that the user wants to access.
- One or more newsgroup names
- One or more e-mail user names or e-mail domain names
- Uniform Resource Locator (URL)

  Specific URLs are used to match the Web page or Web site that the user wants to access. URLs can be specified for an entire Web site or for a particular Web page.

Time Restriction

You can specify <Any>, meaning that the rule always applies, or you can set specific hours of the day and specific days of the week during which the rule is to apply. Use the grid displayed to specify times and days that are appropriate for your network and your users.

Rule Hit Logging

If you select *Enable Rule Hit Logging*, all attempts to connect to destinations and services listed in the access rules are recorded in a log file.

3.1.4 Using Wildcards in Access Rules

Novell BorderManager allows you to enter wildcards (*) in the following source and destination information:

- Hostname
- URL
- E-mail name
- E-mail domain
- News Group

A wildcard is used to skip one or more characters when matching two character strings. For example, the string *.xyz.com matches both www1.xyz.com and www2.xyz.com. Or, the string *xyz* matches any character string that contains the letters xyz, such as abcxyzsd or ?xyz/.

NOTE: There is no wildcard match for eDirectory users, groups, and containers because you cannot manually type eDirectory user, group, or container names into an access rule.

When you use wildcards to match URLs, you will obtain a more specific match if you use // or / to anchor your wildcard name more specifically. For example, to match all URLs that have the letters cgi as part of the path, the best wildcard entry would be */cgi/*. This entry would match all the appropriate URLs, such as http://www.x.com/cgi/xyz. With a less specific use of wildcarding, such as *cgi*, your entry would match http://www.x.com/cgi/xyz, but it would also match http://www.cgi.com/xyz, a URL you did not intend to match.

3.1.5 Access Control Lists

As stated previously, you can configure access rules in NDS or eDirectory Country (C), Organization (O), Organizational Unit (OU), and Server objects. When Novell BorderManager is loaded on a server, it collects the sets of access rules created at each of these eDirectory objects. It first collects rules from its own Server object, then from the Organizational Unit (OU) object above the Server object, the Organization (O) object above the OU object, and finally the Country (C) object above the O object. A Novell BorderManager server’s access control list is simply the collection of all these different sets of access rules in the order given. This consolidated list of rules controls the destinations or services that objects can and cannot access through the Novell BorderManager server and also controls when the objects can access them.

Hierarchical Relationship

The fact that Novell BorderManager allows you to configure and store access rules in different eDirectory objects establishes a hierarchical relationship of access control rules. The location of a given set of access rules in an eDirectory tree defines which servers have those rules built into their access control lists, and in what order.

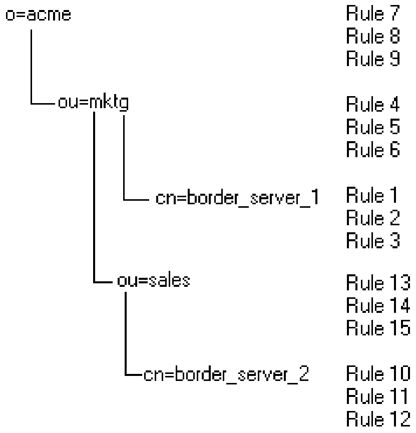
NOTE: The location of an access rule in an eDirectory tree determines which Novell BorderManager servers read the rule and where the rule is placed in a server’s effective rules list. No relationship exists between the context in which a rule exists and the context in which a user exists. A rule defined anywhere in the tree can affect a user defined anywhere else in the tree.

The sequence of each rule list that you create is maintained within the Novell BorderManager server’s access control list. When multiple rule lists are consolidated, rules defined at the server level are placed at the beginning of the access control list; rules defined closer to the root of the eDirectory tree are placed at the end. The sequence of rules in the access control list is important because when an access request is made, Novell BorderManager reads the server’s access control list from the beginning (the server’s rule list) and acts on the first rule that applies to the request.

Example of the Hierarchical Relationship

In this example, CompanyA has the following NDS or eDirectory tree, with different sets of access rules defined in five different places.

Figure 3-1 *Access Control List*



When border_server_1 builds its access control list, it first reads its own access rules, then the access rules in ou=mktg, then the access rules in o=companya. The access rules from the border_server_1 object will be first on the list. The access rules from the o=companya object will be last. The access rules are read from the cn= level up to the top level in sequence. In this example, border_server_1 will *never* read the access rules in either ou=sales or border_server_2.

However, when border_server_2 builds its access control list, it first reads its own access rules, then the access rules in ou=sales, then the access rules in ou=mktg, then the access rules in o=companya. Note that the rules in ou=sales are used only by border_server_2, because it resides under the ou=sales context.

If a user is granted access to both servers, and tries to access the Internet, the access control lists for the two servers would read as shown in the following table:

Table 3-1 *Access Control Lists for two servers*

border_server_1	**border_server_2**
Rule 1	Rule 10
Rule 2	Rule 11
Rule 3	Rule 12
Rule 4	Rule 13
Rule 5	Rule 14
Rule 6	Rule 15
Rule 7	Rule 4

border_server_1	border_server_2
Rule 8	Rule 5
Rule 9	Rule 6
	Rule 7
	Rule 8
	Rule 9

Rule Processing

Again, no relationship exists between the location of a rule and the users or groups that are affected by the rule. The location of an access rule in the eDirectory tree affects only the order in which the access control list is read, and which rules exist in which server's access control list. Although access rules exist in one context, this does not mean that the rules are effective only for users under that context. Any access rule in the access control list of a Novell BorderManager server controls access for all users who request access through that server.

For example, if the first access rule in the border_server_1 access control list is Allow Any, then anybody using that server can access any Web site. If a user under ou=sales uses border_server_1 as the proxy server, that user is allowed to access any Web site.

Novell BorderManager processes access rules in the access control list sequentially to match an access request. When the access request matches an access rule, the action specified in the rule (allow or deny) is immediately applied to the access request and no further processing occurs. If there is no match between the access request and the entire set of access rules (the access control list), the default action (Deny Any) is applied to the access request.

3.2 Implementing Access Control

This section provides more information about configuring access rules, how to build access control lists, and presents several examples. It contains the following subsections:

- Section 3.2.1, "Configuring Access Rules," on page 47
- Section 3.2.2, "Building an Access Control List," on page 48
- Section 3.2.3, "Access Rule Sequence," on page 48
- Section 3.2.4, "Access Rule Example," on page 48
- Section 3.2.5, "Detailed Access Control Example," on page 49

3.2.1 Configuring Access Rules

Novell BorderManager allows you to configure each access rule as either an Allow rule or a Deny rule. Allow rules allow a request to be fulfilled; Deny rules deny the request. You can take two different approaches when you set up your Novell BorderManager access control scheme, as follows:

- In a tightly controlled environment, you might want to configure only Allow rules. If a user's access request fails to match any of the access rules in an entire access control list composed only of Allow rules, the Novell BorderManager denies the default action (deny).

- In a loosely controlled environment, you might want to configure only Deny rules with an Allow Any rule at the end of the access control list. If a user's access request fails to match any of the Deny rules in the access control list, it is allowed by the last Allow Any rule.

3.2.2 Building an Access Control List

As stated previously, Novell BorderManager adds together all the individual access control lists defined at all the different levels of the NDS or eDirectory tree in your network, starting from the Server object, continuing through the container holding the Server object, and concluding at the root of the eDirectory tree.

Within this consolidated list, access control lists defined at the Server object are placed at the beginning, access control lists defined at the root are placed at the end. In other words, rules defined closer to the user are placed closer to the top of the server's access control list. This combined list is the access control list for the Novell BorderManager server, and it is applied to every access request received by the server.

3.2.3 Access Rule Sequence

Within each access control list, Novell BorderManager uses the sequence of access rules in the access control list to determine which rule takes precedence. A rule closer to the top of the list always takes precedence over any rule that follows it in the list. The first rule that matches the access request is the rule that is applied to the access request. If you mix conflicting Allow and Deny rules in an access control list, then you must ensure that the sequence of rules in the list produces the desired effect.

Each time a user makes an access request, such as when a Proxy Services or VPN user initiates a request to access a particular service or destination, Novell BorderManager checks the server's access control list. It searches the list until it finds the first access rule that applies to the requesting object, then it acts immediately on the rule to allow or deny the request. Intelligent sequencing is essential when you create access control rules because Novell BorderManager searches for the first applicable rule in the server's access control list. It does not search for any potentially applicable rules after that.

3.2.4 Access Rule Example

Suppose your company president wants a rule that keeps company employees from accessing the World Wide Web during work hours. You can accomplish this blanket policy with one general access rule at the root of the tree that contains your Novell BorderManager server.

If the vice president of Marketing insists that his people must have access to the Web to perform their jobs effectively, you can satisfy his request with a second rule that you create for the Marketing user group or for specific users within that group.

This approach is one way to implement access rules on your Novell BorderManager server. You use general rules placed higher in the eDirectory tree for far-reaching policies that you want to effect, such as keeping employees from accessing the Web during work hours. You use specific rules placed lower in the NDS or eDirectory tree for more specific cases or exceptions, such as allowing the Marketing group to have access to the Web during business hours.

When two rules conflict with each other (Deny everyone Web access during business hours and Allow members of the Marketing department to access the Web during business hours), the rule closer to the beginning of the server's access control list takes precedence. When no rule is found, the request is denied because the server's default action in the absence of a specific rule is always Deny.

3.2.5 Detailed Access Control Example

In this example, the administrator for XYZ Communications creates the following rules on the XYZ Novell BorderManager server:

Table 3-2 *Access Control Rules*

Rule	Action	Source	Access	Destination
1	Allow	Any	HTTP	www.xyz.com
2	Allow	xyz.com	HTTP	innerweb.xyz.com
3	Allow	exec.xyz.com	Any	Any
4	Deny	xyz.com	Any	www.digitalairlines.com
5	Allow	finance.xyz.com	HTTP	innerweb.xyz.com/prv/finance/*
6	Allow	hr.xyz.com	HTTP	innerweb.xyz.com/prv/hr/*
7	Deny	Any	HTTP	innerweb.xyz.com/prv/*
8	Allow	xyz.com	HTTP	Any
9	Deny	Any	FTP	Any

An analysis of the preceding access rules raises the question of whether the administrator really needs to configure the three Deny rules. Novell BorderManager will automatically deny the user's access request when it reaches the end of the access control list if the request does not match any rule in the list. The answer is it depends on what the administrator wants to accomplish.

An examination of Rule 4, which denies any users in the xyz.com group access to www.digitalairlines.com, and Rule 8, which allows any user in xyz.com to access any destination, reveals that Rule 4 (deny) is necessary.

Rule 7 denies user access to any Web pages inside the innerweb.xyz.com/prv directory after Rules 5 and 6 allow the Finance group and Human Resources group to access their own Web directories.

However, Rule 2 allows any user in xyz.com to access innerweb.xyz.com, which allows access to the pages inside the innerweb.xyz.com/prv directory. Rule 3 allows any user in the exec.xyz.com group to access any destination, which also allows access to pages inside the innerweb.xyz.com/prv directory.

If the administrator does not want users in the xyz.com group to access the innerweb.xyz.com/prv area, then Rule 2 should be moved after Rule 7 (deny). However, if the administrator wants users in the exec.xyz.com group to have access to the innerweb.xyz.com/prv directory, then Rule 7 is not needed.

Rule 8 allows any user in xyz.com to access any destination, which also allows any user to access the host, innerweb.xyz.com. This makes Rule 2 unnecessary. The administrator can also delete Rule

9 because, by default, Novell BorderManager automatically denies user access requests at the end of the access control list when the request does not match any specific rule in the list.

The final access control list looks like the following:

Table 3-3 *Final access control list*

Rule	Action	Source	Access	Destination
1	Allow	Any	HTTP	www.xyz.com
2	Allow	exec.xyz.com	Any	Any
3	Deny	xyz.com	Any	www.digitalairlines.com
4	Allow	finance.xyz.com	HTTP	innerweb.xyz.com/prv/finance/*
5	Allow	hr.xyz.com	HTTP	innerweb.xyz.com/prv/hr/*
6	Deny	Any	HTTP	innerweb.xyz.com/prv/*
7	Allow	xyz.com	HTTP	Any

An analysis of the preceding access control list reveals that any user in xyz.com can access any destination (Rule 7), except the www.digitalairlines.com and innerweb.xyz.com/prv directories (Rule 3 and Rule 6).

Now consider the following requests:

- Can Amy Brentman of the XYZ Finance department use HTTP to connect to www.digitalairlines.com, the Web site of Digital Airlines and the current rival of XYZ Communications?

  If Amy made this request, the access rules would work as follows:

  - Rules 1 and 2 do not apply to Amy's request.
  - Rule 3, however, says that all XYZ employees in all XYZ departments are denied HTTP requests to the Web site of rival Digital Airlines.
  - Rule 7 does allow all XYZ employees in all XYZ departments to use HTTP to connect with any location, but Novell BorderManager already acted on Rule 3, the first rule in the access control list that matched the request.

  The answer is No, Amy cannot access www.digitalairlines.com.

- Jose Lira works in the Human Resources department of XYZ Communications (hr.xyz.com). Can Jose use FTP to copy files from innerweb.xyz.com/prv/hr?

  The access rules would work as follows:

  - The user request is denied because it does not match any rule in the access control list (FTP is not specified in Rules 1 through 7).
  - However, Rule 5 does allow Human Resources (hr) employees like Jose to use HTTP to connect to innerweb.xyz.com/prv/hr.

  Jose can access the site but not he cannot use FTP to copy the files.

- P.V. Singh, President of XYZ, is a member of exec.xyz.com. Can P.V. use FTP to copy files from www.digitalairlines.com?

The access rules would work as follows:

- Rule 2 allows P.V. to use FTP to copy files from www.digitalairlines.com because it allows any member of exec.xyz.com to use any service to connect to any location.
- Rule 3 denies any XYZ employee access to www.digitalairlines.com, but it does not apply because Novell BorderManager has acted on Rule 2.

P.V. can use FTP to copy files from www.digitalairlines.com.

- Roger Rockwell has been in charge of the XYZ Shipping and Receiving department for years. This year, his department was added to the company's intranet. Roger is curious to see what kind of company Web locations are under innerweb.xyz.com/prv.
    - Rules 1 through 5 do not apply to Roger's request.
    - Rule 6 denies Roger's request because it denies all requests by XYZ employees to innerweb.xyz.com/prv.

  Roger cannot see any location under innerweb.xyz.com/prv.

You can also specify periods of the day and days of the week when an access rule is to be in effect, and you can specify that you want all requests against an access rule to be logged.

3.3 Monitoring Access Control

Novell BorderManager logs generate access control log records for Proxy Services. You can view the access attempts for given rules during specified ranges of dates.

NAT Overview and Planning

4

This section describes the decisions you must make before configuring the Network Address Translation (NAT).

This section contains the following information:

4.1 Overview of Circuit-Level Address Translation

To access the Internet, each host must use a globally unique (registered) IP address obtained from an Internet Service Provider (ISP) or from an Internet address registry, such as the Internet Assigned Numbers Authority (IANA). Unless you are requesting a large range of addresses, an ISP should be able to accommodate your addressing needs.

Nevertheless, because it can be costly or impractical to obtain registered IP addresses for every host on your network, you might choose not to assign registered addresses to each host on your private network. Instead you can use NAT, which is a circuit-level solutions provided with the Novell BorderManager® software.

NAT is explained in the following sections:

4.1.1 NAT

NAT is considered a circuit-level solution because it can establish connections to the Internet using registered IP addresses on behalf of multiple hosts on your private network that have not been assigned registered IP addresses. The original circuit (or connection) from a host is terminated at the gateway or NAT interface, and the gateway or NAT interface establishes the actual connection to the Internet for that host. Therefore, multiple hosts can share the same registered IP address if it is assigned to the NAT interface, and the IP addresses of your private network are essentially hidden from the Internet.

NAT does this by translating the private IP addresses to registered IP addresses. NAT enables private clients to access the Internet without the reconfiguration of their private addresses while it hides the addresses of the private network from the Internet.

However, NAT does not require Windows* or a Novell Client™ for Windows. Because NAT operates on a network router interface, the interface's address translation capability can be used by network hosts running any platform, including Windows, Macintosh*, UNIX*, and OS/2*. If these hosts send their TCP/IP packets through the NAT interface, their source IP addresses are not forwarded in the TCP/IP packet headers.

In addition to address translation, NAT can be used to provide other benefits, such as packet filtering based on IP addresses for enhanced network security. When a network interface is configured to use

NAT in any of the three modes of operation, as described in "Selecting a NAT Mode of Operation" on page 54, each TCP/IP packet that reaches the interface is examined for its source or destination IP address. For more information about how NAT filters packets based on source and destination addresses, refer to "Filtering Rules" on page 58.

4.1.2 Whether to Use NAT

NAT might be a choice if the following conditions exist:

- You support other clients using TCP/IP, in addition to Windows clients.
- You are interested in maximizing server performance as it relates to address translation.

  Furthermore, NAT is unaffected by eDirectory problems. NAT checks only for IP addresses in TCP/IP packet headers, its operation does not depend on the availability of eDirectory.

4.2 NAT Configuration Options and Limitations

This section describes the following configuration options:

- Section 4.2.1, "Selecting a NAT Mode of Operation," on page 54
- Section 4.2.2, "Dynamic Only," on page 54
- Section 4.2.3, "Static Only," on page 55
- Section 4.2.4, "Static and Dynamic," on page 55
- Section 4.2.5, "Implementing NAT Modes of Operation," on page 56
- Section 4.2.6, "Considerations for Static Network Address Translation Tables," on page 59
- Section 4.2.7, "Assigning Unregistered Addresses to Hosts Using NAT," on page 60
- Section 4.2.8, "Using Multihoming," on page 60
- Section 4.2.9, "NAT Limitations," on page 61

4.2.1 Selecting a NAT Mode of Operation

NAT can be configured to operate in one of three modes: dynamic only, static only, and a combination of static and dynamic. Dynamic mode is used to allow hosts on your private network, or intranet, to access a public network, such as the Internet. Static mode is used to allow hosts on the public network to access selected hosts on your private network. The combination mode is used when both dynamic mode and static mode functions are required.

The following sections describe each NAT mode of operation and discuss the advantages of using each mode.

4.2.2 Dynamic Only

In dynamic only mode, NAT enables IP hosts on a private network to access the Internet without requiring an administrator to assign a globally unique IP address to each system. Instead, the NAT interface is configured with one public address, and private hosts can then access the Internet through the NAT interface.

Hosts accessing the Internet are dynamically assigned the IP address bound to the NAT interface and a port from a pool of available ports that are constantly reused. Each time a packet is forwarded to the public network, the private address is replaced with the globally unique public address and a randomly assigned port. When the session is completed, the port is returned to the pool to be reassigned as needed. No connections can be initiated from the public network into your private network.

All TCP, UDP, and ICMP packets have their source or destination address (depending on the direction) translated. The public address used for this translation is the primary IP address of the NAT interface, which is specified in the Local IP Address parameter.

NAT provides a pool of 5,000 ports for TCP connections, a pool of 5,000 ports for UDP mappings, and a pool of 5,000 ports for ICMP mappings. To establish a new connection when all 5,000 UDP or ICMP mappings are already used, NAT drops the oldest mapping and provides a port number to the new mapping. To establish a new TCP connection when all 5,000 connections are already used, NAT provides a port number to the new connection by dropping the oldest connection that meets the following criteria in the order shown:

- Any connection that has not transmitted packets for more than eight hours
- Any connection that has been attempting to connect for two minutes but has been unsuccessful (that is, the three-way TCP handshake has not been completed)

4.2.3 Static Only

Static only mode is used for permanent one-to-one mapping of public registered IP addresses to local IP addresses inside a private network. Static address translations are recommended when internal hosts, such as FTP servers or Web servers, are made available to the public network.

In static only mode, NAT is configured with a table of IP address pairs. Each table entry contains a pair of IP addresses for each host that public hosts are permitted to access. The first IP address in each pair is a public IP address to which the private address is mapped; the second address is the address of the host on your private network.

Because public hosts can access private hosts only by using the private hosts' public IP addresses, only those hosts that have their IP addresses defined in the network address translation table are accessible. The NAT interface drops packets addressed to hosts that do not have an address mapping entry in the table. Similarly, to allow private hosts access to the public network using the static only mode, each private host must have its private IP address mapped to a unique public IP address in the network address translation table.

IMPORTANT: When NAT runs in dynamic only mode, a single public IP address and a random port number are assigned to multiple private hosts. When NAT runs in static only mode, all address mappings must be unique. A public address in the network address translation table cannot be mapped to more than one private host.

4.2.4 Static and Dynamic

The combination static and dynamic mode is used if some hosts on your network require dynamic address translation and other hosts require static address translation. For example, your private network might have hosts that you want to access the Internet and might also have resources that you want to be accessed by public hosts. With the combined static and dynamic mode, you can use both methods simultaneously.

To use static and dynamic mode, one public address must be configured for dynamic translations and one public address must be configured for each private host. Because the static and dynamic mode requires more than one public address bound to the same NAT interface, secondary IP addresses (multihoming) must be configured.

You must configure the NAT-enabled interface for multihoming. For more information, see "Using Multihoming" on page 60.

IMPORTANT: When secondary IP addresses are bound to the NAT interface and the static and dynamic mode of operation is selected, the NAT interface automatically uses the primary IP address for dynamic mode. Secondary IP addresses should be mapped to private host IP addresses in the static network address translation table.

4.2.5 Implementing NAT Modes of Operation

The following sections describe how to implement NAT modes of operation:

- "Dynamic Only Example" on page 56
- "Static Only Example" on page 57
- "Filtering Rules" on page 58

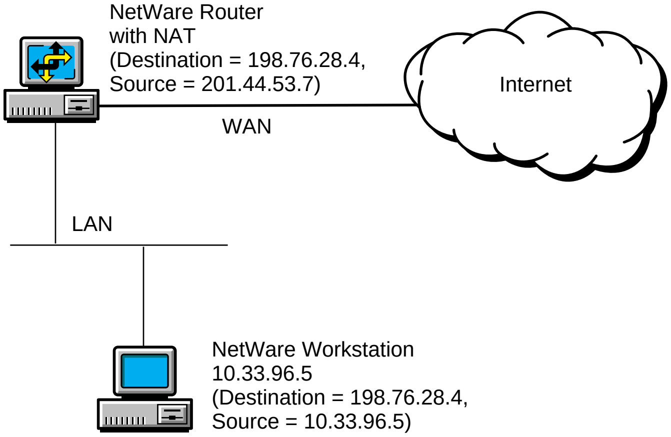
Dynamic Only Example

Figure 4-1 on page 57 shows an application of NAT in dynamic only mode. In the figure, the host on the private network uses the class A address 10.33.96.5. The router's NAT interface to the public network has been configured with the class C address 201.44.53.8. This class C address is globally unique and registered with the Internet Assigned Numbers Authority (IANA) or another Internet registry located outside the United States.

When the host with private address 10.33.96.5 wants to access a host on the Internet with the public address 198.76.28.4, it sends packets to its primary router. The router has a default route configured on the WAN interface, so packets are forwarded to the WAN interface. NAT running on the interface then translates the source address 10.33.96.5 in the IP header to its own globally unique address 201.44.53.8 and assigns a new source port before the packets are forwarded. Similarly, all replying inbound IP packets undergo the reverse address and port translation.

IMPORTANT: The NAT-enabled interface should be configured so that it never uses the Routing Information Protocol (RIP) to advertise the private networks to the public backbone.

Figure 4-1 Dynamic Mode Implementation of NAT



Static Only Example

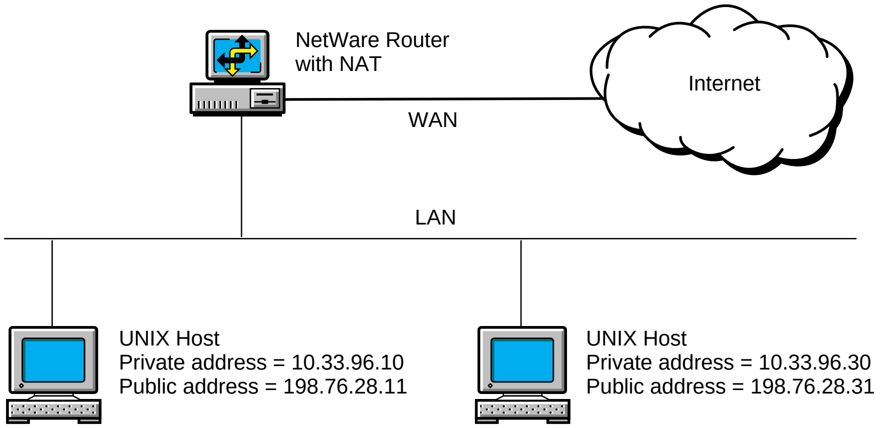
Figure 4-2 on page 58 shows an application of NAT in static only mode. In this case, NAT is configured to allow hosts on the public network to access two UNIX hosts on the private network. The private addresses of the hosts are 10.33.96.10 and 10.33.96.30. The network address translation table is configured to translate these private addresses to the public IP addresses 198.76.28.11 and 198.76.28.31, respectively.

When NAT is configured in this way and packets from public hosts with a destination address of either 198.76.28.11 or 198.76.28.31 are received by the NAT-enabled interface on the NetWare® router, NAT substitutes the destination address of the packets with the appropriate private address and forwards the packets to the private hosts. Reply packets from the private hosts to public hosts undergo the reverse address translation. In this way, hosts on the public network can access specific resources on the private network, but access is limited to only those resources that have their private addresses configured in the network address translation table. A private host whose address is mapped to a public address in the network address translation table can also access any public host.

When NAT is used in static mode with a multiaccess configuration, the public router must have a static host route for each address pair defined in the NAT static mapping table. If NAT is used with a numbered point-to-point configuration, you are not required to configure static host routes.

IMPORTANT: The NAT-enabled interface should be configured so that it never uses the Routing Information Protocol (RIP) to advertise the private networks to the public backbone.

Figure 4-2 Static Mode Implementation of NAT


NetWare Router
with NAT
Internet
WAN
LAN
UNIX Host
Private address = 10.33.96.10
Public address = 198.76.28.11
UNIX Host
Private address = 10.33.96.30
Public address = 198.76.28.31

Filtering Rules

The types of packets that the NAT interface filters are largely determined by the mode in which NAT is operating. The NAT mode is set using the Status parameter. There are four possible settings for this parameter: Disabled, Dynamic Only, Static and Dynamic, and Static Only. For more information on how to configure NAT parameters, see *Novell BorderManager 3.9 Administration Guide*.

- "Disabled" on page 58
- "Dynamic Only" on page 58
- "Static and Dynamic" on page 59
- "Static Only" on page 59

Disabled

If a NAT-enabled interface is configured as Disabled, all incoming and outgoing packets are passed without any modifications to either the source or destination IP address or port. This is the default setting.

Dynamic Only

If a NAT-enabled interface is configured as Dynamic Only, the filtering rules are as follows:

- Packets that originate from the private network or from services running on the NetWare server have the source address and port translated and are forwarded to the destination address.
- Inbound ICMP packets of types 0, 3, 4, 8, 11, 12, 17, and 18 are allowed access. All other types of ICMP packets, including ICMP redirect (type 5), are dropped. Inbound ping request (ICMP echo) packets are answered by NAT when requests are addressed to the NAT interface IP address.
- Packets that originate from the public network and do not correspond to requests that originated from the private network are dropped.

NOTE: NAT translates any outbound packets that pass through the interface. If a private network has both registered and unregistered IP addresses, the registered IP addresses are translated to the registered address configured for the NAT interface.

Static and Dynamic

If a NAT-enabled interface is configured as Static and Dynamic, the filtering rules are as follows:

- Inbound packets that are not destined for one of the public addresses configured in the network address translation table or that are not translatable are dropped. Untranslatable packets are those that cannot be matched with an existing outbound dynamic flow.
- Outbound packets from any private hosts are translated. Packets from configured static private hosts are treated according to the rules for static mode, and all other packets are treated according to the rules for dynamic mode.

Static Only

If a NAT-enabled interface is configured for Static Only, the filtering rules are as follows:

- Only packets received from the public network with a destination address that matches one of the public addresses configured in the network address translation table are allowed to access the private network.
- Only the private hosts whose addresses are specified in the network address translation table are allowed to access the public network. Any packets from other private hosts are dropped.
- Packets that originate from the public network and that are not destined to any public addresses configured in the network address translation table are dropped.

NOTE: By configuring filters for a NAT-enabled interface, a secure static translation can be created by allowing only specified services, hosts, or networks access from the public network.

For more information about configuring filters, refer to the packet filtering online documentation.

4.2.6 Considerations for Static Network Address Translation Tables

Consider the following when you configure address translation mappings in a static network address translation table:

- When using NAT with packet filtering, you must modify the filters to account for the address translations configured in the network address translation table. Because filtering operations are performed before address translations, the inbound filters must permit untranslated addresses to reach the NAT interface and the outbound filters must permit translated addresses to be routed through the NAT interface.
- When NAT is used in static mode with a multiaccess configuration, the public router must have a static host route for each address pair defined in the NAT static mapping table. If NAT is used with a numbered point-to-point configuration, you are not required to configure static host routes.
- Although the private and public addresses of each static network address translation table entry technically cannot be the same IP address, one exception is required if other TCP/IP services are accessed from the public address bound to the NAT-enabled interface. For example, if the NAT interface’s public address is also used to access an FTP or Web server, the static network address translation table must have an entry with the public address mapped to itself.

- If static mode is desired, NAT cannot be enabled on more than one LAN or WAN interface that reaches the same private host. For static network address translation, only one route per host is allowed.
- Using static mode with a static network address translation table is not practical if the TCP/IP Bindings setting for *Remote Router Will Dynamically Assign IP Address* is set to Yes. Thid is because, the assigned public address is subject to change. If this setting is configured in the INETCFG utility, only dynamic mode can be used.

4.2.7 Assigning Unregistered Addresses to Hosts Using NAT

To determine which IP address to assign to private hosts when NAT is used, use the guidelines in RFC 1918.

In summary, RFC 1918 explains that the Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of IP space for private Internets:

10.0.0.0 to 10.255.255.255 (10/8 prefix)[lnbrk]172.16.0.0 to 172.31.255.255 (172.16/12 prefix)[lnbrk]192.168.0.0 to 192.168.255.255 (192.168/16 prefix)

The first block is referred to as a 24-bit block, the second block as a 20-bit block, and the third block as a 16-bit block. Note that the first block is a single class A network number, the second block is a set of 16 contiguous class B network numbers, and the third block is a set of 256 contiguous class C network numbers. Because the backbone routers of the Internet have filters that prevent them from forwarding packets to these network addresses, using the addresses offers additional protection for private hosts hidden by the NAT in the event that the gateway, NAT, or firewall malfunctions or is configured incorrectly. However, the routers used by some ISPs might not have filters for these addresses, thereby allowing access to your private hosts by any IP hosts outside your network that use the same ISP.

An enterprise can use the network numbers of the address space described in RFC 1918 without any coordination with IANA or an Internet registry. Therefore, the network numbers can be used by many enterprises. Addresses within this private address space must be unique within the enterprise, or within the set of enterprises that choose to share the address space in order to communicate with each other using their private internetwork.

4.2.8 Using Multihoming

Multihoming is when multiple IP addresses on the same network are bound to a single network interface. IP addresses other than the first address bound to the network interface are referred to as secondary IP addresses.

The most common use of secondary IP addresses on the same network interface is for a single Web server to operate as though it were several Web servers. A different secondary IP address can point to a different Web page on the same Web server, depending on the DNS domain name that is used to reach the server.

Multihoming is commonly used with NAT running in static mode, proxy services, and Virtual Private Networks (VPNs). In all cases, the secondary IP addresses are configured on a network interface that already has a primary IP address bound to it.

When multiple interfaces are configured on a server, the secondary address is associated with the interface that has the same network address bound to it; that is, the network portions of the two IP addresses match. If you attempt to configure a secondary address that is not valid on any of the networks bound to existing interfaces, the address is rejected and an error message appears on the server console.

When multihoming is used with NAT, proxy services, or VPNs, the secondary addresses must be configured manually. For more information, see *Novell BorderManager 3.9 Administration Guide*.

4.2.9 NAT Limitations

NAT has the following limitations:

- Because different TCP/IP applications can embed and use IP addresses uniquely, NAT does not support applications that embed an IP address in the data portion of the TCP/IP packet. However, FTP is an exception to this rule. NAT performs special processing to allow FTP to function properly. For more information about this limitation, refer to RFC 1631.
- Multicast and broadcast packets are not translated.

Proxy Services Overview and Planning

5

Novell® BorderManager® 3.9 Proxy Services handles all requests for remote Internet documents and other objects from browser clients within the firewall. The proxy server makes the remote connection and transparently transfers information to the clients.

This section contains overview and planning information for Novell BorderManager Proxy Services. It also includes sample setup information to help you plan your configuration. This chapter contains the following sections:

- Section 5.1, "Overview of Proxy Services," on page 63
- Section 5.2, "Functionality of Forward Proxy Caching," on page 67
- Section 5.3, "Application Proxies," on page 70
- Section 5.4, "Additional Proxy Services Features," on page 77
- Section 5.5, "Designing and Planning Proxy Services," on page 81

5.1 Overview of Proxy Services

The growth and increased popularity of the World Wide Web has created a corresponding growth in network traffic. With this growth have come delays, slower response times, and security concerns.

The network traffic problems are partly due to the repeated retrieving of objects from remote Web servers on the Internet. Novell BorderManager Proxy Services can help improve performance by locally caching frequently requested Internet information. In general, Proxy Services stores copies of frequently requested Web information closer to the user, thereby reducing the number of times the same information is accessed over an Internet connection, the download time, and the load on the remote server.

This section contains the following information:

- Section 5.1.1, "Types of Caching," on page 63
- Section 5.1.2, "Interaction with Other BorderManager Services," on page 64
- Section 5.1.3, "Proxy Technology," on page 65
- Section 5.1.4, "Supported Protocols," on page 65
- Section 5.1.5, "Proxy Services Benefits," on page 66
- Section 5.1.6, "Proxy Services Features," on page 66

5.1.1 Types of Caching

There are four types of caching:

- "Passive Caching" on page 64
- "Active Caching" on page 64
- "Negative Caching" on page 64

- "Hierarchical Caching" on page 64

Passive Caching

With passive caching (also called basic or on-demand caching), the client (browser) sends a request directly to a proxy server, which is an HTTP server that usually runs on a firewall server. The proxy server locates the object in its cache and returns the object to the client. If the object is not in the cache, the proxy retrieves a copy from the origin Web server on the Internet, stores it in the cache on the proxy server, and returns a copy of the object to the client. The object is cached for a preset period of time or until the cache is full. If the cache disk space is low, older objects are removed from the cache. Subsequent browser requests for the cached object are made to the proxy server at local intranet speeds. This reduces Internet traffic and the request load on the source Web server, thereby reducing the delays in returning information to the client.

To the client, the proxy server has the same basic functionality as the Web server (with a subtle difference in submitting requests). To the Web server, the proxy server has the same basic functionality as the client. The proxy builds its cache based on the Web sites that users visit. When an object is retrieved from the Web and put in a cache, a Time-To-Live (TTL) value is associated with the object. Before the TTL expires, requests are filled from the cache for that object. When the TTL expires, the Web server is contacted for a newer version, the update is stored in the cache, and a new TTL is calculated.

Active Caching

Active caching is an add-on to passive caching to improve performance. With active caching, the proxy automatically sends a request to the origin server to retrieve an object. The server updates objects that are more frequently accessed or requested, have longer TTLs, and are actively cached during periods of low server load.

Negative Caching

Negative caching occurs when a proxy attempts to resolve a request for a URL that does not exist or cannot be located or accessed. In this case, the proxy caches the negative result so that future requests for that URL are resolved quickly. The proxy continues to check in the background and refreshes the cache when the pages become available. Negative caching occurs for HTTP error conditions such as 403 (forbidden request) and 404 (URL not found).

Hierarchical Caching

Hierarchical caching allows information to be retrieved from the nearby or closest proxy servers instead of from the originating Web server. HTTP and FTP acceleration (reverse proxy cache acceleration) also allows static information to be cached by and retrieved from the border proxy servers instead of the origin Web servers to reduce the Web server load. The proxy cache uses cache aging information that Web servers provide to browsers to determine how long pages should be cached.

5.1.2 Interaction with Other BorderManager Services

Access control is used by the Proxy Services software applications to forward and filter connections for such services as HTTP, Gopher, and FTP. The host running Proxy Services is known as the gateway. In general, Proxy Services allows services only for which there are proxies. For example, if a gateway has proxies for FTP, then only FTP can be requested; requests for all other services are ignored.

With gateways, you can hide the names and addresses of internal systems—the gateway is the only hostname known outside the system. Also, traffic can be logged before it reaches the internal hosts. Proxy Services improves security by hiding private network domain names and addresses and sending all requests through a single gateway. For more information about gateways, refer to Chapter 4, “NAT Overview and Planning,” on page 53.

5.1.3 Proxy Technology

Proxy Services is based on both the first-generation CERN proxy technology and the newer, second-generation Harvest/Squid hierarchical proxy cache technology. The Harvest/Squid technology enhances standard CERN proxy cache services with negative URL caching and negative domain Name System (DNS) caching, and introduces hierarchical caching through the Internet Cache Protocol (ICP).

The Harvest project, an Internet Resource Discovery Project contract performed by the University of Colorado, introduced ICP hierarchical caching to improve Internet Web performance and scalability. The project was transferred to the National Laboratory for Applied Network Research (NLANR) in early 1996 as the basis for the Squid project. The goal of the Squid project is to facilitate the evolution of an efficient national architecture for handling highly popular information.

5.1.4 Supported Protocols

Novell BorderManager Proxy Services supports the following protocols and applications:

- HTTP (0.9, 1.0, and 1.1), including HTTPS support and Secure Sockets Layer (SSL)
- FTP
- Domain Name System (DNS)
- Gopher
- Simple Mail Transfer Protocol/Post Office Protocol 3 (SMTP/POP3)
- Network News Transfer Protocol (NNTP)
- RealAudio and RealVideo*
- Real Time Streaming Protocol (RTSP)
- SOCKS 4 and 5
- Generic TCP/UDP
- HTTP Transparent proxy
- Telnet Transparent proxy

The passive mode (PASV) is supported for FTP to allow the firewall administrator to deny incoming connections above port 1023, if necessary. Otherwise, normal (PORT) FTP mode is used. Proxy Services also supports the HTTP protocol over the Internetwork Packet Exchange™ (IPX™) software. Novell IPX/IP clients, as well as other clients, can directly access the proxy server using the gateway client transparent proxy feature.

5.1.5 Proxy Services Benefits

Novell BorderManager Proxy Services combines an Internet proxy, a Web caching facility, and the NDS™ or Novell eDirectory® software to provide World Wide Web access from within a firewall. Proxy Services has the following benefits:

- Reduces WAN traffic to the Internet and on the primary Web server by providing local LAN access to cached information. Proxy Services also reduces the load on Web Internet servers and increases Internet and intranet performance.
- Uses a single protocol on the LAN (for HTTP proxy only). Users do not need to have separate clients; HTTP is used to communicate with a proxy server. The proxy server uses the appropriate protocol; FTP, Gopher, and so on for HTTP requests to access documents from the network.
- Improves intranet security by hiding the local network from the Internet. Private network domain names and addresses are hidden and all requests are sent through a single gateway. This applies to forward proxy only. Reverse proxy is used to hide the origin host from the client or local network.
- Enhances intranet security with access control and content filtering.
- Distributes LAN client requests across multiple proxy servers, for example, FTP requests on one server and HTTP requests on another server.
- Reduces the disk space requirements for retrieved information on client workstations and reduces the load on Web Internet servers.
- Enables document access even when the Internet or intranet Web server is down or inaccessible, if the document is already cached by the proxy server and Time-To-Live is not expired.
- Undeletes and serves if the origin server is down.
- Provides a single point administration based on eDirectory.
- Logs and filters client transactions.

These benefits apply to both Internet and intranet Web sites. Because Proxy Services supports open Internet standards, it can be used with Novell intranet and Internet products, as well as with other vendors' browsers and Web servers.

5.1.6 Proxy Services Features

Proxy Services includes the following features:

- Support for HTTP (0.9, 1.0, 1.1), FTP, Gopher, DNS, and SSL clients
- Hierarchical caching based on the Internet Cache Protocol (ICP) and other protocols
- HTTP and FTP server accelerator (reverse proxy)
- Application proxies, including SMTP proxy, NNTP proxy, DNS proxy, SOCKS, HTTP Transparent proxy, Telnet Transparent proxy, and RealAudio and RTSP proxies
- SOCKS client support
- Batch downloading of URLs
- Content filtering for Java*
- Simple Network Management Protocol (SNMP) Management Information Base (MIB)

- Access control lists based on eDirectory user identity, IP addresses, domains, and URLs
- Windows-based management console and configuration
- SurfControl (third-party site-blocking software)
- Event logging in Text and Relational Database Management System (RDBMS) formats

5.2 Functionality of Forward Proxy Caching

This section describes how forward proxy caching works, and includes information on caching hierarchies, object types and caching, and proxy server security.

- Section 5.2.1, “Forward Proxy Caching,” on page 67
- Section 5.2.2, “Caching Hierarchies,” on page 68
- Section 5.2.3, “Object Types and Caching,” on page 68
- Section 5.2.4, “Proxy Servers and Security,” on page 68

5.2.1 Forward Proxy Caching

Proxy Services acts as an intermediary between hosts on a protected network and the Internet or intranet, or between Internet clients and servers on your network. When a user requests an Internet service, such as HTTP, the client submits the request to the proxy, which then acts on the client’s behalf. The proxy checks its local cache for the data and, if the data is available, sends it to the client immediately. If the data is not available, the proxy requests the data from the hierarchical cache servers or the origin server on the Internet, and then returns the data to the client.

The proxy server works as both a client and a server. As a server, it receives requests from intranet clients. As a client, it forwards the requests to the origin Internet server.

When a client makes a regular HTTP request (without using a proxy), the HTTP server receives only the path and keyword portion of the requested URL. For example, the user enters the following command:

```
http://host.com/marketing/doc.html
```

The browser sends the following command to host.com:

```
GET /marketing/doc.htm
```

In this example, the protocol specifier http and the hostname are already known to the remote HTTP server. The requested path specifies the object available from that server.

When a client sends a request to a proxy server through a browser, the proxy uses HTTP and the GET method. The client (browser) uses HTTP when communicating with the proxy, even when accessing an object on a remote server that uses a different protocol, such as Gopher or FTP. If a different protocol is being used, the proxy identifies the protocol and uses that protocol and the full URL to make the request. The proxy server has all the information necessary to make the request to the remote server.

For example, for an FTP protocol request, the proxy uses the following command:

```
GET ftp://host.com/marketing/doc.html
```

All the information is used when requesting information from the origin server. The proxy requests the document using FTP, the results are returned to the proxy server as an FTP reply, and the server then sends the information to the user as an HTTP reply.

5.2.2 Caching Hierarchies

You can set up a hierarchy of proxy cache servers to reduce the WAN load and resolve requests. Whenever a request for an object cannot be resolved, the proxy server contacts its neighbors (peers) and parents using the Internet Cache Protocol (ICP), a simple resolution protocol. The proxies exchange queries and replies to gather information and select the best location from which to retrieve a requested object.

If the URL matches a listing on a configurable list of substrings, the object is retrieved directly from the origin server rather than from other proxy servers. If the request is a cachable object, the proxy server sends the request to the siblings and parents using UDP broadcast. The object is retrieved from the closest available site. Caching hierarchies reduce the load on origin Web servers and distribute the load across many cache servers. See "ICP Hierarchical Caching" on page 78 for more information about hierarchical caching and how it works.

5.2.3 Object Types and Caching

Not all objects can or should be cached. Some types of objects are of no value when cached because they change too frequently. Other types of objects require authentication before they can be accessed.

HTTP supports the HEAD method to retrieve only the header to determine how recent an object is. If an object has not been modified since the time specified in a header request, the object is not returned and the cached object is used.

HTTP also supports the If-Modified-Since request header, enabling a conditional GET request. The GET request contains the date and time the object in the proxy cache was last modified. If the object has been modified since the stored date and time, a new copy is retrieved.

Usually, the proxy server does not cache the following types of objects:

- Objects that are password-protected
- Objects with /cgi-bin/ or ? in their URLs
- Objects that are larger than a preconfigured size
- Objects associated with protocols other than HTTP, FTP, or Gopher
- Any HTTP header that contains Pragma:no-cache, Cache-Control: Private, Set Cookie, WWW-Authenticate, or Cache-Control:no-cache

You can specify additional noncachable object types. For more information, see *Novell BorderManager 3.9 Administration Guide*.

The Novell BorderManager proxy uses the cache aging information that Web servers usually provide to browsers. This information specifies how long pages should be cached. The HTML text is typically only a small part of the transmitted data, even for sites that dynamically generate HTML pages. The majority of the data consists of images that are static and cachable. To improve performance, you can fine-tune cache aging policies.

5.2.4 Proxy Servers and Security

Proxy Services interacts with the following to provide additional proxy server security:

- "Access Control" on page 69

- "Novell Internet Gateway Clients" on page 70

One benefit of establishing proxy servers on your intranet is to increase security through access control and the logging of URL requests. Proxy servers have two types of security:

- **Outbound Security:** The proxy server can restrict access by URL site to Internet protocols, such as HTTP, FTP, and Gopher. For example, you can restrict access to Web sites that do not fit your company policy or are not essential to completing company work.
- **Inbound Security:** The proxy server keeps internal network addresses secure by hiding client IP addresses from the Internet and substituting these addresses when requesting information from Web servers.

The proxy server provides tighter security than using only address filtering. The proxy server determines the address of a packet and the entire context of the session in which the packet is being sent, making it easier to identify suspicious packets.

The proxy server can be used as a part of a firewall solution or together with firewall solutions from other vendors. It can be used in front of, within, or behind existing firewalls.

Access Control

For additional security, access is controlled using access control list rules. You can set up Proxy Services access control to do the following:

- Control user access to intranet servers with sensitive data
- Deny user access to certain unproductive Web sites
- Restrict the use of unauthorized or inessential applications
- Restrict incoming and outgoing mail based on username or domain
- Deny access to news groups

For example, you might want to deny access to Web sites that do not fit your company policy or are not essential to completing company work.

With access control lists, the proxy server restricts access based on the source and destination IP addresses, URLs, domains, and NDS or eDirectory usernames. The proxy server also works with other third-party site-blocking software, such as SurfControl, to block sites by category.

The access control list is a set of rules that either allow or deny a specific action. The access control list rules are stored in the eDirectory database.The access control list module checks the HTTP request and determines whether any of the access rules apply. If a rule applies, the specified action is performed. Otherwise, the default rule is applied. You can create access control rules at the Country, Organization, Organizational Unit, and Server object levels. Rules can be based on criteria such as users, groups, IP addresses, or services. For more information, see *Novell BorderManager 3.9 Administration Guide*.

With NDS or eDirectory, access control lists are associated by container, group, user, or server. Access control lists can apply to all proxies in an organization, thereby giving management a global view.

URL restrictions can also be based on usernames. In addition, HTTP/IPX and HTTP/IP clients can directly access proxy servers using the gateway client transparent proxy feature.

Novell Internet Gateway Clients

The proxy server also supports the HTTP protocol over IP (as well as any WinSock-based program). Novell IPX/IP clients can directly access the proxy server. When you configure your browser on the gateway client to go through a proxy, the gateway client automatically detects the proxy servers using NDS or eDirectory. The gateway client then redirects the requests to the proxy.

5.3 Application Proxies

This section describes in detail the following supported application proxies:

- Section 5.3.1, "HTTP Proxy," on page 70
- Section 5.3.2, "Blocking Virus Requests in an HTTP Accelerator," on page 71
- Section 5.3.3, "FTP Proxy," on page 72
- Section 5.3.4, "Mail (SMTP/POP3) Proxy," on page 74
- Section 5.3.5, "RealAudio and RTSP Proxies," on page 74
- Section 5.3.6, "DNS Proxy," on page 75
- Section 5.3.7, "HTTPS Proxy," on page 75
- Section 5.3.8, "SOCKS Client," on page 75
- Section 5.3.9, "Generic Proxy," on page 76
- Section 5.3.10, "Transparent Proxy for HTTP," on page 76

5.3.1 HTTP Proxy

There are two types of HTTP proxy:

- "HTTP or Forward Proxy" on page 70
- "HTTP Accelerator or Reverse Proxy" on page 71

HTTP or Forward Proxy

HTTP proxy resolves URL requests on behalf of Web clients on your network. This is also known as forward proxy. These requests are cached, if possible, on the proxy server to increase the speed of delivering the same content the next time the same information is requested.

HTTP itself is an application-level protocol used for distributed, collaborative, hypermedia information systems. It is generic and allows systems to be created independently of the data being sent. It is also an object-oriented protocol that can be used for name servers, distributed object management systems, and so on. HTTP servers use HTTP as the primary application protocol, allowing users to access and exchange Web files. The HTTP protocol can also be used for communication between users, proxies, gateways, and other Internet protocols, such as SMTP, NNTP, FTP, and Gopher.

HTTP communication is usually over TCP/IP connections, on default port 80, although other ports can be used.

HTTP Accelerator or Reverse Proxy

The proxy server can be configured as an HTTP accelerator to protect an intranet server from the Internet and reduce the load on the public Web servers maintained on the intranet. HTTP acceleration, also known as reverse proxy cache acceleration or Web server acceleration, creates a front-end processor to a Web server. An HTTP accelerator server lies between one or more Web servers and the Internet and represents the Web servers to any clients accessing them. An HTTP accelerator can also be used to create a local mirror site of a remote server.

When the Internet user queries DNS for the Web server address, it returns the address of the requested Web server. The HTTP accelerator listens for HTTP requests on port 80 (or another configured port) and processes all incoming Web requests. Requests for objects that can be cached—static information that does not change often, such as HTML pages and GIF images—are processed by the proxy. Requests for objects that cannot be cached—dynamic information that changes frequently—are processed by the origin Web server on port 80. In general, approximately 90 percent of a typical Web server content is static and 10 percent is dynamic.

You can set up an HTTP accelerator server to retrieve information or references to cachable objects from a Web server and cache the information on a Novell BorderManager server. This reduces loading on the Web server. The HTTP accelerator server forwards only requests and references that are not in the cache to the Web server.

If your site receives requests for a high percentage of objects that can be cached, the HTTP accelerator reduces the Web load. For even greater performance, you can cache objects of a more volatile nature, such as stock quotes, and specify an accuracy delay time to users.

Novell BorderManager reverse proxy can handle more TCP connections than an origin Web server (typically UNIX or Windows NT*).

HTTP acceleration has the following benefits:

- Provides caching for Web servers
- Reduces the load on the Web servers and speeds them up
- Protects Web servers
- Protects IP networks in conjunction with the other Novell BorderManager services

5.3.2 Blocking Virus Requests in an HTTP Accelerator

In the past few months, we have seen an increase in self-propagating malicious viruses such as Code Red and Nimda, which are designed to inflict maximum damage to computer systems around the world. The methods used by this new breed of viruses demonstrate the growing sophistication of virus and worm attacks.

For example, in the case of Code Red, the worm attempts to connect to a certain TCP port on a randomly chosen host, assuming that a Web server will be found. Upon successful connection to the host, the attacking computer sends an HTTP GET request that attempts to exploit a known vulnerability in Microsoft* Internet Information Server (IIS) Web servers. If the exploit is successful, the worm begins executing on the victim host. Depending on the day of the month, it either attempts to further propagate itself by connecting to other randomly chosen IP addresses, or it launches a packet- flooding "denial of service" attack against a fixed IP address. The ultimate goal of this type of attack is to generate so much illegal traffic to the site that service is denied to the site’s legitimate users.

Although these viruses infect only certain vulnerable Web servers and routers that do not use NetWare, their method of attack can impact the performance of Novell BorderManager Proxy Servers that are used to accelerate these Web servers. To protect against such attacks from the Novell BorderManager Proxy side, there must be some mechanism in place to examine all incoming HTTP requests and reject those that are identified as coming from virus-infected computers. The main problem lies in differentiating between legal and illegal requests and acting accordingly, without adversely affecting the performance of the Novell BorderManager Proxy Server.

To accomplish this, Novell has added a Virus Pattern Recognition and protection enhancement to Novell BorderManager. This enhancement includes features to facilitate its configuration and monitoring.

This section provides an overview of the functionality of the Virus Pattern Recognition feature as a mechanism for protecting Web servers against distributed denial-of-service (DDoS) attacks. The solution involves creating a database of known virus patterns. The Novell BorderManager Proxy Server then compares every incoming request with the existing pattern database, and blocks any request that perfectly matches one of the patterns in the database.

The main goals in the design of this feature were:

- To make it easy to add and delete virus request patterns in the database.
- To allow the pattern database to be updated "on the fly", without bringing down the Proxy Server.
- To provide automatic detection of changes in virus patterns and subsequent updating of the pattern database.
- To offer effective console-based monitoring.
- To minimize the impact on performance.

In discussing the functionality of this feature, it is helpful to understand the following terminology used to categorize HTTP requests:

- **Suspect Request:** Any request that is suspected to generate from a virus- infected client or server.
- **Virus Request:** Any request that is determined to generate from a virus-affected client or server.
- **Humble Request:** Any request that is valid to the origin Web server (a non-virus request).

5.3.3 FTP Proxy

There are two types of FTP proxy:

- “FTP Proxy” on page 72
- “FTP Reverse Proxy” on page 73

FTP Proxy

FTP is the standard Internet protocol used for file transfer. FTP proxy is used to proxy FTP requests when users use pure FTP clients, for example, the LAN WorkPlace® software, UNIX, Macintosh, and so on.

FTP proxy has the following benefits:

- Centralized access control
- Data caching for FTP data files
- Ability to resume data file transfer after temporary loss of connection to FTP server
- Anonymous users allowed
- URL representation of FTP data

Standard FTP requires a user account on the server being accessed. Anonymous FTP does not require a user account and provides access to specific files on the Internet. The username is anonymous or ftp.

You can use proxy servers to control access to authenticated FTP sites. When an FTP proxy server is placed on a firewall, all FTP client requests in the intranet must pass through the FTP proxy server. This helps enforce centralized control over Internet access and scans data that is being sent or retrieved by users within an organization.

The FTP intranet client (or user) must first connect to the FTP proxy server by entering the IP address or name of the proxy server, for example, ftp://novell.com. The user must then enter the following to identify the origin host and connect to the FTP proxy:

```
USER   ProxyUserName$  DestFTPUserName$ DestFTPHostName

PASS   UsereDirectoryPassword$  DestFTPPassword
```

where *ProxyUserName* is the NDS or eDirectory username, *DestFTPUserName* is the FTP username on the destination server, *DestFTPHostName* is the hostname or IP address of the destination FTP server, *UsereDirectoryPassword* is the user's eDirectory password, and *DestFTPPassword* is the user password on the destination server. Only the FTP hostname *DestFTPHostName* is required. If the *DestFTPUserName* is missing, it is assumed to be anonymous, and no password is required. The *ProxyUserName* is required only if FTP authentication is enabled. The proxy makes the final connection to the origin host or server.

Both active and passive FTP modes are supported, and can be enabled or disabled. Active mode (PORT) posts a listener on the intranet and allows clients to make a connection to the intranet machine, a less secure method. Passive mode (PASV) for FTP allows the client to initiate the connection to a remote FTP server. PASV mode is supported to allow the firewall administrator to deny incoming connections above port 1023, if necessary.

FTP Reverse Proxy

FTP reverse proxy, or FTP accelerator, is an application that is placed in front of the FTP server. The FTP accelerator acts as an FTP server to Internet users and protects the FTP servers behind the firewall from outside break-ins. The FTP accelerator scans inbound and outbound data, and with third-party support, can trap any viruses being sent through the system.

The FTP accelerator also caches frequently requested data and FTP files for anonymous users and helps accelerate FTP requests. This process is useful because most FTP requests from the Internet are from anonymous FTP users. Caching shifts the load from FTP servers to the reverse FTP proxy.

5.3.4 Mail (SMTP/POP3) Proxy

Electronic mail is the most fundamental and useful of Internet services. It is also the most vulnerable. To create a secure environment, you must be able to restrict access to outside mail to only a few machines, screen messages for hostile applets or scripts, and avoid other malicious e-mail schemes.

SMTP handles electronic mail exchange between mail servers, accepting mail and sending it directly to the destination mail domains or delivering it to an intermediate relay agent. Post Office Protocol 3 (POP3) is used to handle the user electronic mailboxes on servers.

The Mail proxy server provides secure SMTP mail services for incoming and outgoing mail. SMTP allows intranet users to send mail to the Internet in a secure manner. Similarly, Internet users can send mail through SMTP to intranet users in a secure manner. Incoming mail is scanned for viruses, filtered for junk mail, and controlled using access control lists.

SMTP proxy can perform the following access control and filtering for outgoing and incoming mail:

- Enforce access control based on usernames and mail domain names for incoming and outgoing mail.
- Hide internal mail domain names and usernames. The Mail proxy can be configured to overwrite the From address so that only the primary mail domain name for an organization is exposed to the Internet.
- Filter for Multimedia Internet Mail Extensions (MIME). Mail is scanned and filtered for attachments, including non-ASCII character sets, nontext data, rich text messages (with formatted text), and multipart messages.
- Scan and filter incoming e-mail for viruses and junk mail. Unwanted junk mail is scanned using access control lists that combine mail domain filtering and content filtering.

Mail proxy can be used in an organization between the existing intranet mail server and the Internet, or between the intranet and the Internet without an existing intranet mail server. The following e-mail commands are allowed by the Mail proxy: HELO, MAIL, RCPT, DATA, RSET, HELP, NOOP, and QUIT.

5.3.5 RealAudio and RTSP Proxies

Using the RealAudio and RTSP proxies, a RealAudio player communicates with a RealAudio server to play back audio or video as it is downloaded (as opposed to downloading an entire program before hearing it). RealAudio and RTSP eliminate the delays that can occur during download, especially with slower modems. They also support several quality levels and nonaudio features such as HTML pages synchronized with voice.

The RealAudio and RTSP proxies allow players inside the firewall to connect to the specified proxy, which then connects to the requested RealAudio server outside the firewall. The proxies hide any intranet RealAudio servers that should not be visible to the Internet. No caching is performed. You can configure reverse proxy if any RealAudio or RTSP servers should be visible to the Internet. RealAudio proxy requires RealPlayer* 2.0 or later, which can be configured with the hostname and port number used by the proxy.

The RealAudio player and server can use one of the following methods of communication:

- **TCP Only:** In this mode, a single full-duplex TCP connection is used for both control and audio data delivery between the player and the server. The standard TCP connection port on the server is 7070.
- **Standard UDP:** In this mode, the player sets up two network connections with the server. A full-duplex TCP connection is used for control and negotiation. A one-way UDP path from the server to the player is used for audio data delivery.
- **Robust UDP (optional):** In this mode, the player sets up three network connections with the server. A full-duplex TCP connection is used for control and negotiation. A one-way UDP path from the server and the player is used for audio data delivery. A second one-way UDP path from the player to the server is used to request that the server resend lost UDP audio data packets.

5.3.6 DNS Proxy

DNS is a distributed data system that translates hostnames to IP addresses and vice versa. DNS also stores and accesses other information about hosts.

When enabled, the DNS proxy acts as a DNS server for clients on the intranet. A listener is posted on the DNS port. When a DNS request is received from a client, the DNS proxy checks its local DNS cache and returns a response, if available. If the address is not in the cache, the DNS proxy forwards the request to the configured DNS name servers. The proxy caches only the responses of Internet class and Internet address queries.

The client must have the private IP address of the DNS proxy configured as the address of its DNS server.

On the server, you can set up the IP addresses of the DNS name servers and the domain name in the `sys:\etc\resolv.cfg` file.

5.3.7 HTTPS Proxy

The HTTPS proxy provides the ability to access secure sites using SSL over a persistent IP connection. The browser sends an HTTPS request as an SSL request through the proxy, which then tunnels the request to the origin Web server.

5.3.8 SOCKS Client

This feature enables a proxy to authenticate through a SOCKS 5 firewall. This release also supports the forwarding of HTTP traffic only.

SOCKS is a circuit-level gateway protocol. With SOCKS, hosts behind a firewall can gain full access to the Internet without full IP support. When SOCKS support is enabled, all requests sent to the Internet are forwarded to a SOCKS 5 server when the proxy is used for caching only.

When the proxy receives a request, it checks its cache. If the requested object is not in the cache, the proxy makes a TCP connection to the SOCKS server and redirects the request from the intranet to the SOCKS server, allowing for more secure Internet access. The SOCKS server then connects to the origin server and retrieves the object. The proxy simply acts as a SOCKS client to the SOCKS server and is used for caching only. Null (no username or password) and username/password authentication are supported.

This release requires that the proxy server and the SOCKS server are both on the same intranet. The reason is that in the username/password combination, SOCKS authentication uses clear text to send the password.

5.3.9 Generic Proxy

A generic proxy is a circuit-level, pass-through proxy used to serve multiple protocols when an application proxy is not available. A mapping is created between the address and ports, creating a tunnel to the destination host. When the generic proxy server receives a connection request from the intranet, it forwards the request to the mapped address, connects to it, and transfers data between the two connections.

To establish connections using TCP services for which there is no application proxy, a generic TCP proxy should be set up at the proxy server. You can also define a generic UDP proxy. When connecting to the proxy, the user is connected to the internal host. Authentication is available for a generic TCP proxy. A user must be authenticated using access control list rules before connecting to a remote host. Authentication is not available for a generic UDP proxy.

You can apply access control rules to a generic TCP proxy. Access can be allowed or denied based on the following:

- The IP address or hostname of the original host
- The port number associated with the origin host
- The IP address of the source host in the intranet

5.3.10 Transparent Proxy for HTTP

There are two types of transparent proxies:

- "HTTP Transparent Proxy" on page 76
- "Telnet Transparent Proxy" on page 77

HTTP Transparent Proxy

A transparent proxy can be implemented for HTTP using either of the following features:

- On the server using the HTTP Transparent proxy feature
- On the client using the gateway client transparent proxy feature

An HTTP Transparent proxy enables users to use their Web browsers without reconfiguring each browser to point to a proxy. This feature is useful if you have limited time and cannot immediately reconfigure the browsers for all your users. It is also useful when you want to enforce network security and ensure that all client requests pass through a proxy.

The HTTP Transparent proxy intercepts traffic between the client and the origin Web server, and funnels it to a proxy server. Relative URLs are translated to absolute URLs. For HTTP Transparent proxy only, traffic from a configurable list of ports or IP addresses is intercepted. Only the ports or addresses on the list participate in forwarding traffic to the proxy.

To use HTTP Transparent proxy, you must ensure that all HTTP requests are sent through the proxy server. Therefore, the proxy server must be the default router or provide the only access to the Internet.

Telnet Transparent Proxy

A transparent proxy can be implemented for Telnet using the Telnet Transparent proxy.

A Telnet Transparent proxy enables users to use their Telnet application without reconfiguring their applications to point to a proxy. This feature is useful when you want to enforce network security and ensure that all client requests pass through a proxy.

The Telnet Transparent proxy intercepts traffic between the client and the origin Telnet server and funnels it to a proxy server. For the Telnet Transparent proxy only, traffic from a configurable list of ports is intercepted. Only the ports on the list participate in forwarding traffic to the proxy.

To use a Telnet Transparent proxy, you must ensure that all Telnet requests are sent through the proxy server. Therefore, the proxy server must be the default router, be in the routing path, or provide the only access to the Internet. IP forwarding must be enabled on the server.

5.4 Additional Proxy Services Features

This section contains the following subsections describing additional features supported by Proxy Services:

- Section 5.4.1, “Batch Downloading,” on page 77
- Section 5.4.2, “Java Content Filtering,” on page 77
- Section 5.4.3, “Proxy Authentication Using SSL,” on page 78
- Section 5.4.4, “ICP Hierarchical Caching,” on page 78

5.4.1 Batch Downloading

You can schedule downloads of HTML files from a Web site to the local cache. You can download one URL, multiple URLs up to a specified number of links, or an entire Web site. You can specify batch downloads for both forward and reverse HTTP proxies. However, reverse proxy does not download links that are external to a site.

Schedule a download before the workday starts to optimize your use of network resources. Using batch downloading keeps the cache of objects up-to-date for users.

5.4.2 Java Content Filtering

HTML documents can have Java applet tags embedded in them without your knowledge.

A Java applet is a Java program that can be included in an HTML page. When you use a Java-compatible browser to view a page that contains a Java applet, the applet’s code is transferred to your system and executed by the browser.

Once downloaded, the applet can excessively consume system resources, interfere with other applets, inspect and change client files, and make unauthorized client connections. With Novell BorderManager, you can enable Java blocking and filter the received HTML pages for any embedded applets. Any applets are removed from the document before it enters the system.

5.4.3 Proxy Authentication Using SSL

Proxy authentication to a proxy server can be accomplished in two ways:

- "Single Sign-On Authentication" on page 78
- "Proxy Authentication Using SSL" on page 78

If proxy forward or reverse authentication is enabled and both single sign-on and SSL are enabled, the proxy server first tries to authenticate the user through single sign-on. If the single sign-on attempt fails, the proxy server establishes an SSL connection with the client and then authenticates the user with an NDS or eDirectory username and password.

Single Sign-On Authentication

When you use Novell Client32, single sign-on eliminates the need for additional proxy authentication after you log in to NDS or eDirectory.

When the client generates a browser request, the proxy server verifies whether the client is authenticated. If the client is not authenticated, the proxy server requests that the client initiate a background authentication to the proxy server. All the protocol exchanges occur in the background, and the user is not prompted to enter an additional username and password.

Single sign-on is successful only when the client machine is running the Novell Client 32 software and has logged in to NDS or eDirectory. The client machine must also be running `dwntrust.exe` and `clntrust.exe`. These files are located in the `sys:public` directory on the server.

Single sign-on occurs on port 3024 on the server. If single sign-on has been enabled on the same Novell BorderManager server for Proxy Services, only one background authentication is required for a user to use both services. This is because of the shared port on the server. For single sign-on to work, packet filtering firewalls in the routing path between a gateway or proxy client and a Novell BorderManager server must allow packets designated for port 3024 to pass through.

Proxy Authentication Using SSL

SSL authentication also eliminates the need for an additional proxy password. For clients running Novell Client 32, SSL authentication is used when single sign-on fails or is not enabled. For non-Novell clients, SSL is the primary method for eliminating the need to send an eDirectory username and password in clear text.

SSL ensures private and secure communication between the browser and the proxy server using a public-private key encryption system. When the client makes a browser request, the proxy server responds by requesting authentication using a Java applet or an HTML form. In response, the browser creates an SSL connection and sends the eDirectory username and password, encrypted over SSL, to the proxy server. The proxy server then authenticates the user with eDirectory.

To use SSL authentication, you must generate a certificate for the proxy server, which is then used for setting up encrypted channels.

5.4.4 ICP Hierarchical Caching

ICP hierarchical caching includes the following elements:

- "Cache Hierarchies" on page 79

- “Multicast Groups” on page 80
- “Source Roundtrip Time” on page 80
- “ICP Domain Routing” on page 80
- “ICP Access Control” on page 80

Proxy servers maintain knowledge about each other by using intelligent cache topologies. Hierarchical proxy caching increases performance by retrieving first-time access and negative cache data from the optimal nearby proxy server without retrieving this data from the origin Web server.

You can configure hierarchical cache routing to improve the performance of local cache misses. Two cache routing systems are provided: first-generation CERN cascading and second-generation Internet Cache Protocol (ICP) hierarchical caching.

The CERN cascading system provides standard HTTP forwarding through proxy chains. The ICP hierarchical cache system provides advanced cache routing through a designed cache topology. ICP provides maximum performance, scalability, and fault tolerance for both intranet and Internet Web caching.

ICP hierarchies are built on neighborhoods of proxy cache services. These neighborhoods are made up of parents and peers. The local proxy cache service normally has five neighbors. You can define multiple parents to improve performance and fault tolerance. ICP dynamically determines optimal cache fulfillment in the neighborhood. It also automatically detects down neighbors and recovers them.

The difference between peers and parents is in the fulfillment of URL objects that are not cached in the neighborhood. Only parents can be requested to obtain a URL for the neighborhood. The basic ICP fulfillment strategy is to obtain a neighborhood cached object from the most optimal neighbor (the one that responds first with a cache hit), and to obtain missed URL objects for the neighborhood and its local cache from the most optimal parent (the parent that responds first with the most preferred cache route).

The following are some basic guidelines for ICP neighborhood topology:

- Limit the number of neighbors to five. More than five neighbors will flood the network with ICP requests without improving the quality of service.
- Choose parents that are en route to the intranet or Internet (for example, organizational, connected to the firewall or ISP, and so on).
- Choose two or more parents for fault tolerance. Establish a routing preference with parents based on the quality of service (bandwidth, load balancing, and so on).
- Choose peers that have close proximity relationships (high bandwidth and low latency).

Cache Hierarchies

ICP is a UDP-based message format used for communication among proxy caches. ICP is used in a cache mesh to locate cached Web objects in neighboring caches. Caches exchange ICP queries and replies to select the best location from which to retrieve an object. After the location is determined, the object is retrieved using an HTTP proxy.

When a query is received, the cache first checks its local cache, then sends ICP queries to its neighbors. The neighbors return ICP replies indicating a hit or miss. Depending on the replies, the proxy might access the parent neighbor to retrieve the object from the origin server. Novell BorderManager is not set up to send queries through a Web hierarchy by default.

Multicast Groups

Neighbors can be configured as multicast groups. A multicast group is a list of addresses on which the ICP server receives multicast IP queries. A request for information can be sent to a multicast group.

The ICP server is configured with a list of the multicast groups or addresses that will participate in the ICP requests. The ICP client is configured with the responder list, a list of all acceptable neighbors (a unicast list) that can respond to a multicast query. This allows the ICP client to verify whether the responses are from a valid neighbor. The client keeps track of both the multicast responders and the multicast neighbors.

Setting up multicast groups reduces traffic congestion and configuration time. With multicast groups, you are not required to configure each neighbor separately, and the neighbors do not send multiple packets.

Source Roundtrip Time

If a cache miss occurs, the proxy uses the source roundtrip time to determine whether to send the request to a parent or to the origin server. To calculate the roundtrip time, the proxy sends an Internet Control Message Protocol (ICMP) query to the origin server and the parent caches. The proxy uses the route that returns the shortest roundtrip time.

ICP Domain Routing

ICP domain routing is used to regulate ICP traffic based on host domains. The geographic distribution of neighbor caches might dictate that some domains can be better served by one cache over others. When you set up ICP clients, you can associate each neighbor with a list of route domains that it will serve. This increases response time and efficiency and reduces network traffic.

For example, suppose the following configuration is in effect:

```
Neighbor host1 parent http_port=8080 icp_port=3130 .com .net .au .de
Neighbor host2 parent http_port=8080 icp_port=3130 .edu .mil
Neighbor host3 parent http_port=8080 icp_port=3130
```

In this example, assume the queries for root domains .com, .net, .au, and .de are forwarded to host1 or host3. Queries for .edu and .mil are forwarded to host2 or host3. Queries for all other root domains are forwarded to host3. ICP queries are not sent to a neighbor if it does not serve the requested URL host domain.

The default configuration is null, or all neighbors receive all queries. You can configure neighbors to process queries for one or more domains on the domain list.

ICP Access Control

ICP access control is configured on the ICP server and is used to verify whether proxies are allowed to send a request. You can set up a list of allowed clients (or clients that can send the ICP request). ICP access control is separate from Novell BorderManager access control.

5.5 Designing and Planning Proxy Services

This section contains examples of how you can design various applications of the Novell BorderManager Proxy Services.

- Section 5.5.1, "Web Client, Server, and Network Acceleration," on page 81
- Section 5.5.2, "Proxy Application Examples," on page 89

5.5.1 Web Client, Server, and Network Acceleration

This section describes the three primary ways to use proxy caching:

- Web client acceleration (standard proxy cache)
- Web server acceleration (reverse proxy cache acceleration or HTTP acceleration)
- Network acceleration (ICP hierarchical caching)

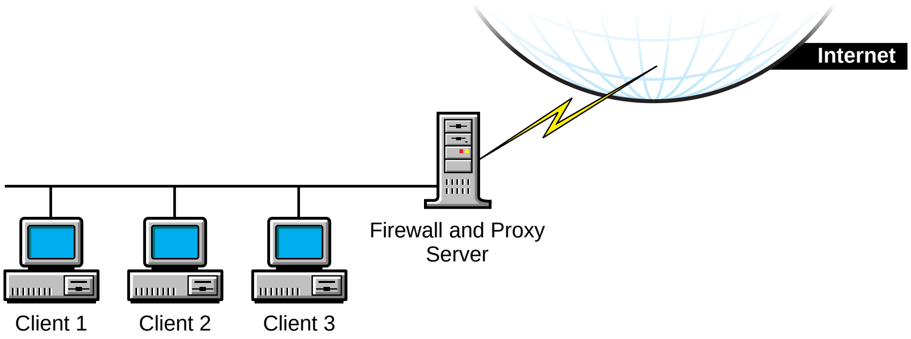
This section also provides several examples of how you can use caching. In these examples, CompanyA is implementing several proxy cache solutions to enhance its enterprise network: client acceleration, server acceleration, and network acceleration. For each type of caching, examples are given for both intranet and Internet use.

Web Client Acceleration (Standard Proxy Cache)

In Web client acceleration, the proxy server is located between clients and the Internet, as shown in Figure 5-1 on page 81. The proxy server intercepts requests from clients for Web pages and supplies the requested pages to the client, if cached, at LAN speed. This eliminates the delay that occurs when the origin Web site is accessed and minimizes the traffic between the corporate network and the Internet.

The proxy server makes requests to Web servers for the intranet clients, using appropriate protocols such as HTTP, FTP, and Gopher. The proxy server caches URLs, HTML pages, and FTP files to accelerate subsequent requests to the same objects.

Figure 5-1 *Client Accelerator Configuration*



Identifying Cache Sites

When planning the implementation of proxy servers and caching on your network, you must identify which sites would benefit from caching. Look for the following when identifying client acceleration sites:

- Sites with multiple clients

  In almost all cases, operating these clients through a proxy server dramatically improves performance. This is especially true if groups of employees are accessing the same Internet or intranet Web sites, thereby increasing the probability of cache hits. Caching also uses resources more efficiently, including intranet Web servers.
- Sites that require control over client access to the Internet

  You can control access to the Internet by establishing easy-to-understand access control rules.

Internet Client Acceleration Example

For example, suppose that Company A wants to give its employees access to the wealth of information available on the Internet. However, the company also wants to restrict access only to those Internet Web sites that contribute to the workplace. This results in two requirements:

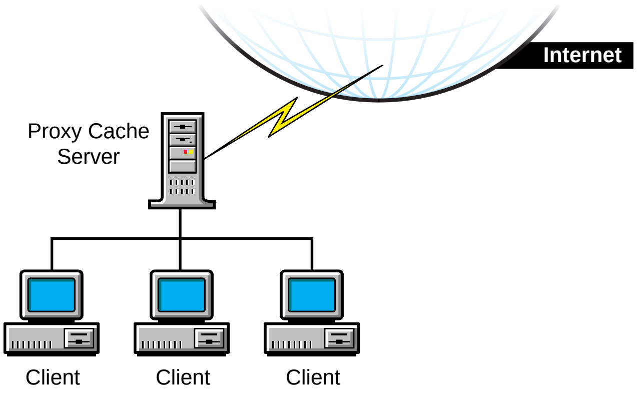
- Restrict Internet access

  The company must apply a consistent and manageable Internet access policy to all employees, for example, to deny access to certain Web sites. Because many employees travel extensively, access control must be implemented globally, regardless of the employee location.
- Accelerate Web access

  The company must reduce the time employees spend waiting for Web pages to load while still giving them full access to the information they need.

To meet both requirements, Company A implements proxy servers as client accelerators in all of its facilities, using access control list rules established in NDS or eDirectory by the network administrator. All employee Web browsers are configured to operate through proxy servers. The proxy cache servers greatly accelerate Web page loading and permit control over Internet access. This configuration is shown in Figure 5-2 on page 82.

Figure 5-2 *Internet Client Acceleration Configuration Example*



Intranet Client Acceleration Example

Various groups within Company A have published extensively on internal Web sites. Some of the published information is company public, or accessible by all employees. Other information is privileged, or accessible only by employees who have a need to know. For example, some advanced development information is available only to certain engineering or management groups. The published information is spread across a large number of internal Web sites.

Company A has two requirements for intranet Web site access by employees:

- Restrict access to privileged information

  A set of rules must be implemented that specify the intranet Web information that can be accessed. Because many employees travel extensively, access control must be implemented globally, independent of employee location. Otherwise, access management becomes complex.

- Reduce network load

  The Company A intranet consists of a number of sites that are interconnected by WAN links. Because of the high bandwidth requirements of Web access, particularly those sites with extensive graphics, the transfer of Web pages over WAN links must be minimized.

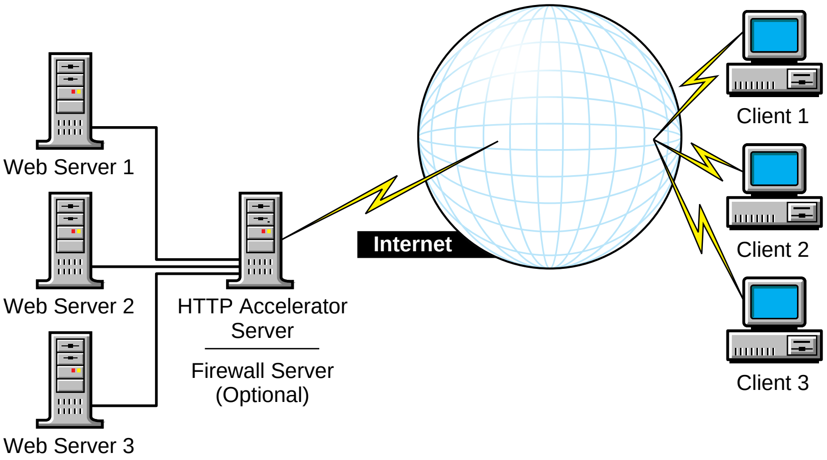
Because of the flexibility of Proxy Services, the same proxy servers used to restrict Internet Web access in the first example, "Web Client Acceleration (Standard Proxy Cache)" on page 81, can also be used to restrict access to intranet Web sites. In addition to storing the Internet access restrictions in eDirectory, the administrator stores intranet Web access rules. This approach gives the administrator centralized and global control of both Internet and intranet access from a single point, greatly simplifying access management.

Web Server Acceleration (HTTP Acceleration)

With Web server, or HTTP, acceleration, the proxy server acts as a front end to one or more Web servers and caches all information that belongs to the Web server, as shown in Figure 5-3 on page 83. When a client requests information from a Web server, the request is diverted to the proxy server. The proxy server supplies the cached pages to the client at high speed. This method accelerates access and takes the request load off the publishing Web servers, allowing them to handle publishing and dynamic content more efficiently.

Proxy Services can provide acceleration for all popular Web servers in any combination.

Figure 5-3 *Web Server Accelerator Configuration*



Identifying Cache Sites

When planning the implementation of proxy servers and caching on your network, you must identify which sites would benefit from caching. Look for the following when identifying server acceleration sites:

- Internet or intranet Web servers that have a high level of usage

  You can improve performance and capacity significantly by using proxy servers. (Refer to "Internet Server Acceleration Example" on page 84.)

- Intranet Web servers that contain both company public and company privileged information

  Representing these Web servers on the network with a proxy server simplifies access management, tightens security, and increases performance. (Refer to "Intranet Server Acceleration Example" on page 85.)

- Sites with a variety of Internet or intranet Web server platforms

  Representing these Web servers on the network with a proxy server consolidates and centralizes access management, tightens security, and increases performance. (Refer to "Intranet Server Acceleration Example" on page 85.)

Internet Server Acceleration Example

The public Web site of Company A, http://www.ACo.com, receives millions of hits daily from a worldwide audience. The site was previously serviced by multiple Web servers. Recently, the company set up several proxy servers to serve as front ends to the Web servers, as shown in Figure 5-4 on page 85. This approach provides three important benefits:

- Increased capacity

  Company A can expand the content of its Web site and accommodate more site visitors without upgrading hardware. In fact, each proxy server—an economical 200-MHz Pentium* Pro machine with 128 MB of RAM and a 16-GB disk—can handle approximately 250 million hits every 24 hours. Estimating conservatively that only 50 percent of all hits are cached, each proxy server can take 125 million hits every 24 hours. This greatly reduces the load on the Web servers and might reduce the number of Web servers required as well. In this example, one proxy server is sufficient to handle the load for the foreseeable future. However, Company A installed multiple proxy servers for a fault-tolerant solution.

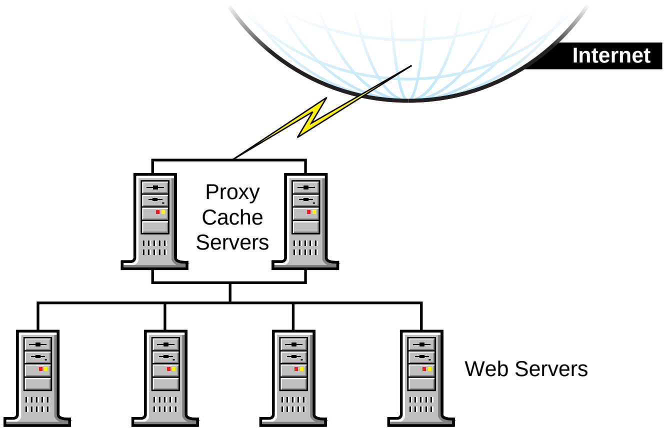
- Increased performance

  Because of the significant performance boost provided by caching, Web site visitors download pages faster, making their experience more satisfying.

- Enhanced security

  The proxy servers isolate the Company A Web servers from the Internet, protecting them against unauthorized access.

Figure 5-4 *Internet Server Accelerator Configuration Example*



Intranet Server Acceleration Example

Many of the groups in Company A publish information on internal Web servers on the company's intranet. These servers are scattered around the world and are accessed by employees who are also located around the world. Unlike the information on the public Internet Web site of Company A, much of the information published internally is sensitive and access to it must be restricted. Complicating this situation is that the information resides on a variety of Web server platforms, including NetWare, UNIX Apache, Netscape, and NCSA*, making access management complex and difficult.

The Company A solved the problem by creating front ends to its intranet Web servers with proxy servers at each site. For example, at its headquarters, the company installed 10 proxy servers as front ends to the 50 intranet Web servers at that site, as shown in Figure 5-5 on page 86. Access control was transferred from the Web servers to the proxy servers. This approach results in the following benefits:

- Effective and consistent increased security

  The proxy servers isolate the Web servers from the network, increasing their resistance to unauthorized access. By moving security to the proxy servers, Company A can provide the same strong security across all Web servers, regardless of the Web server platform. This makes the security policy easy to implement.

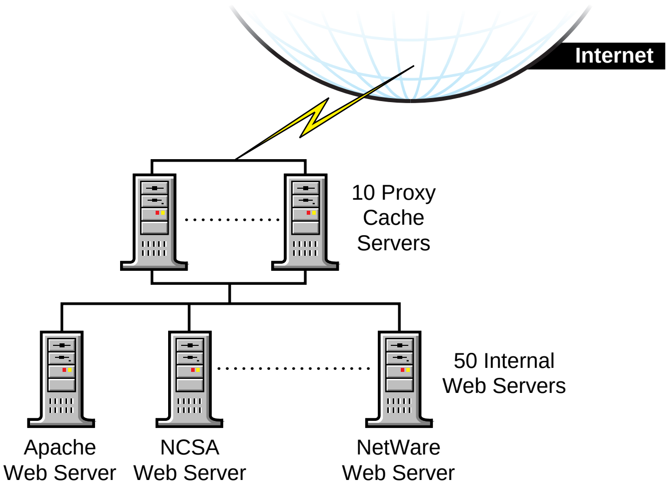
- Centralized and simplified access control

  Access control is implemented through access control rules stored in eDirectory. As a result, an administrator can manage security for all servers from a single point, regardless of the server platform. This greatly simplifies access management. In addition, because access control is implemented through NDS or eDirectory, it is independent of employee location, and the same access control rules are applied no matter where a user logs in. The access control list used for server access control in this example is synchronized with the access control list used for client access control in "Web Server Acceleration (HTTP Acceleration)" on page 83, ensuring uniform access control in both client and server acceleration within the intranet.

- Increased performance

  The proxy servers increase the speed of Web page access. Employees receive the information they need faster, becoming more productive.

Figure 5-5 Intranet Server Accelerator Configuration Example

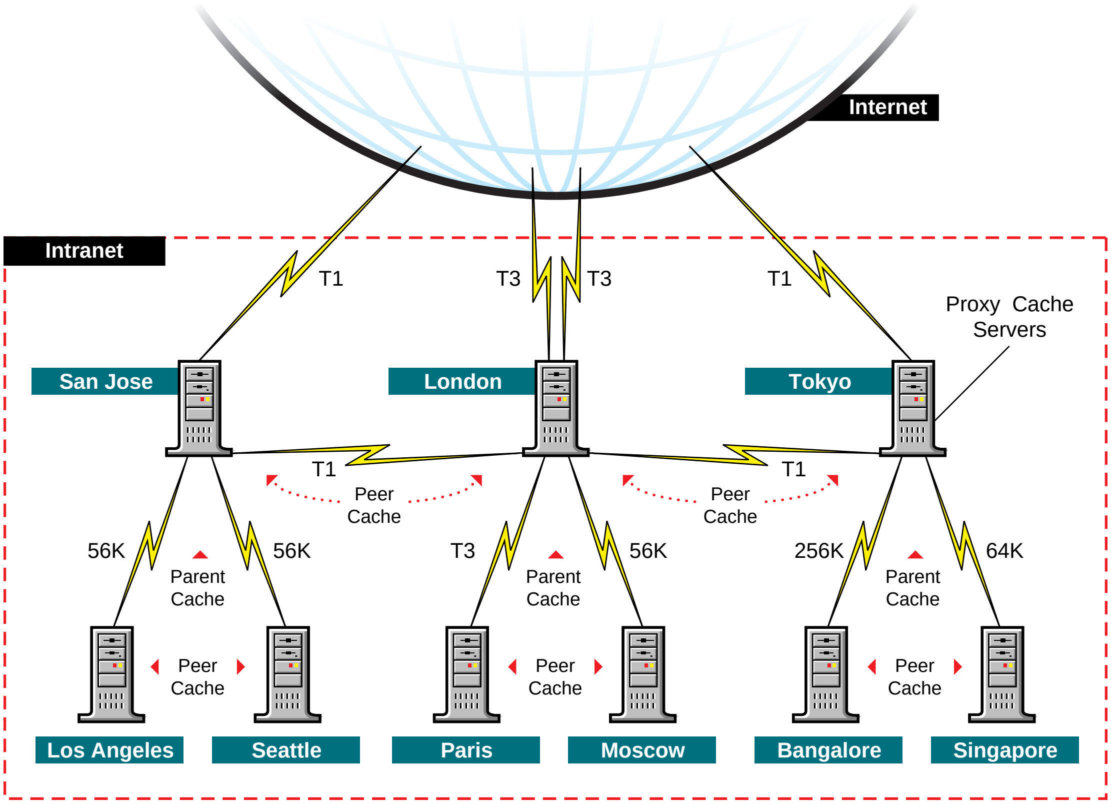


Network Acceleration (ICP Hierarchical Caching)

With network acceleration, or ICP hierarchical caching, multiple proxy servers are configured in a hierarchical, or mesh, topology, as shown in Figure 5-6 on page 87. The proxy servers are connected in a parent, child, or peer relationship. When a miss occurs, the proxy contacts the other servers in the mesh to find the requested cached information. The nearest proxy cache that has the requested information forwards it to the requesting proxy server, which in turn forwards it to the requesting client.

ICP hierarchical caching reduces the WAN traffic load and increases valuable bandwidth. In addition, because the requested information is sent from the nearest proxy server, network delays are minimized. This reduces user wait times and increases user productivity.

Figure 5-6 *Network Accelerator Configuration*



Identifying Cache Sites

When planning the implementation of proxy servers and caching on your network, you must identify which sites would benefit from caching. Look for the following when identifying network acceleration sites:

- Sites with slow links to the Internet

  Use hierarchical caching to deliver information to clients at either LAN speeds or over high-speed intranet WAN links.

- Sites with multiple LANs

  Use multiple proxy servers to partition LAN traffic. For example, you can install a proxy server in each building. This approach reduces backbone traffic, that is, the traffic between the LANs. It also uses your resources more efficiently, accommodates more usage over the same backbone, and speeds up backbones that have become sluggish because of increased traffic.

- Congestion and delay problems at LAN points within WANs

  A hierarchical mesh of proxy servers can increase the available bandwidth of your WAN by reducing WAN traffic. For example, a company has three sites: a field sales office, a regional office, and corporate headquarters. If a client at the field sales office needs a Web page from corporate headquarters, the client would access the page directly from the Web server and prevent other clients from using two WAN links. However, if the page is cached at the regional office, the client would use only one WAN link, thereby reducing traffic on the other link.

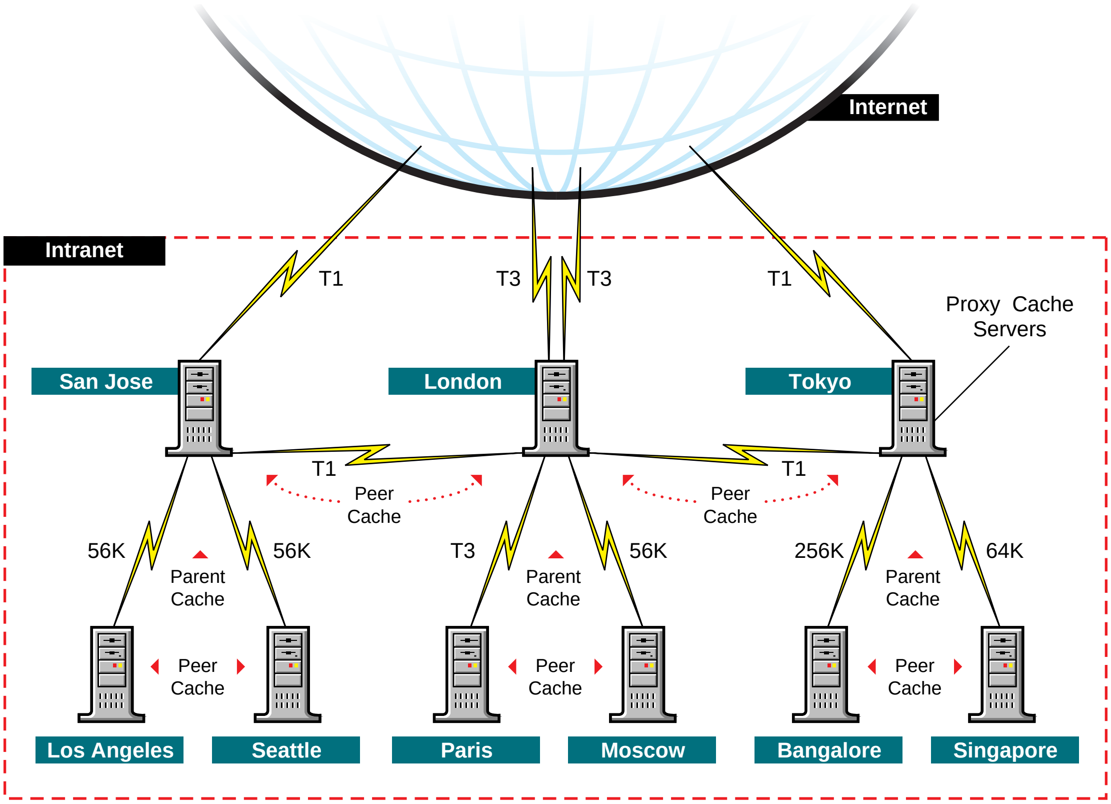
Intranet Network Acceleration Example

Company A is a large organization with worldwide facilities. As a result, employees and Web servers are widely scattered. Employees must have easy and fast access to internal Web information, regardless of their location or the location of the target Web server. In addition, because of the high cost of network equipment and the even higher cost of managing it, the company must obtain the highest utilization possible from its network resources.

Company A implemented a hierarchical mesh of proxy servers, as shown in Figure 5-7 on page 88. Hierarchical caching reduces the load on Web servers and reduces WAN traffic by allowing clients to access cached intranet Web information from the closest proxy server.

For example, a Los Angeles-based employee might be in Paris and need to access information from a Web site in Los Angeles. Although no one at the Paris office has recently accessed that information, an employee in the London office has, and the information is cached on the proxy server in London. Instead of routing the client's request all the way to Los Angeles, the proxy server in Paris can access the information from the proxy server in London. This reduces network delay and eliminates slower, more expensive transatlantic traffic on the network.

Figure 5-7 *Intranet Network Accelerator Configuration Example*



Internet Network Acceleration

Just as a hierarchical mesh of proxy servers can be used to accelerate intranet performance, it can be used on a much larger scale to accelerate Internet performance. The National Laboratory for Applied Network Research (NLANR) is working on such a project.

According to a recent NLANR report, the Internet's sustained explosive growth calls for an architected solution to the problem of scalable wide area information dissemination. While increasing network bandwidths helps, the rapidly growing populace will continue to outstrip network and server capacity as they attempt to access widely popular pools of data throughout the network. The need for more efficient bandwidth and server utilization transcends any single protocol such as FTP, HTTP, or whatever next becomes popular.

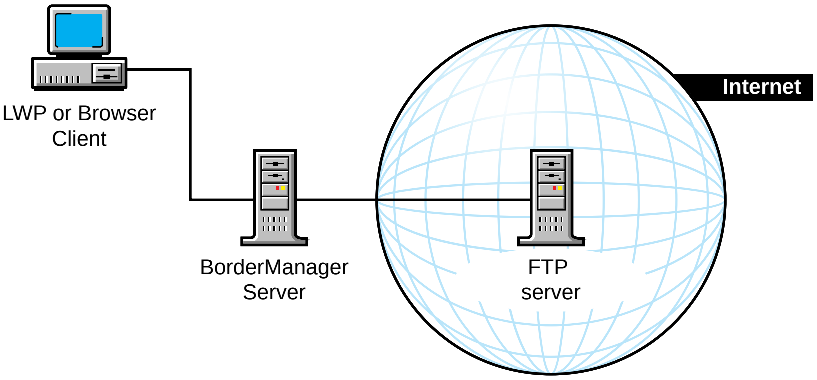
The basic Internet client-server model (in which clients connect directly to servers) is wasteful of resources, especially for highly popular information. There are many examples in which server systems have not been able to cope with the demands placed upon them for popular information.

5.5.2 Proxy Application Examples

This section contains examples of FTP, FTP reverse proxy, Mail (SMTP), DNS proxy applications, and an example of SOCKS.

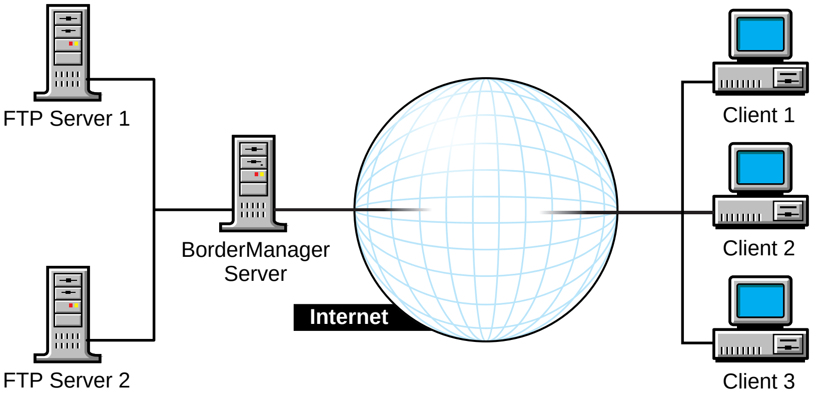
The Figure 5-8 on page 89 shows an example of FTP acceleration using a Novell BorderManager proxy server on the firewall. The browser client can access the FTP server through the proxy server.

Figure 5-8 *FTP Acceleration*



The Figure 5-9 on page 90 shows an example of FTP reverse acceleration. In this example, the client accesses the two FTP servers on the intranet through the Novell BorderManager proxy server on the firewall.

Figure 5-9 *FTP Reverse Acceleration*



The following two figures show two examples of using the Novell BorderManager Mail proxy to connect to an external mail server. The Figure 5-10 on page 90 shows an example of a small company without an internal mail server. The Novell BorderManager proxy server acts as a mail server, handling all SMTP and POP3 requests from the intranet and the corresponding mail from the external mail server on the Internet. The Figure 5-11 on page 91 shows a larger company with its own internal mail server. The internal mail server uses the Novell BorderManager proxy server to exchange mail with outside or public mail servers.

Figure 5-10 *Mail Proxy without an Internal Mail Server*

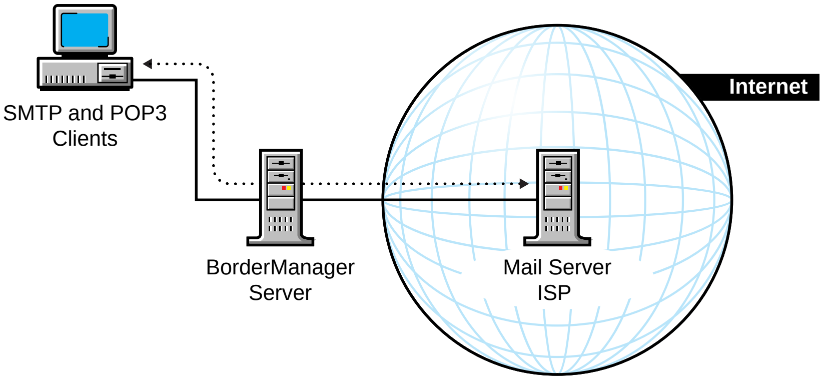
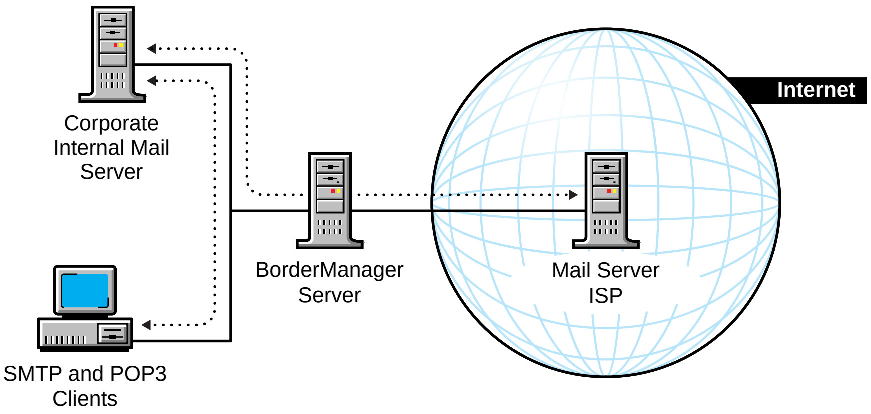
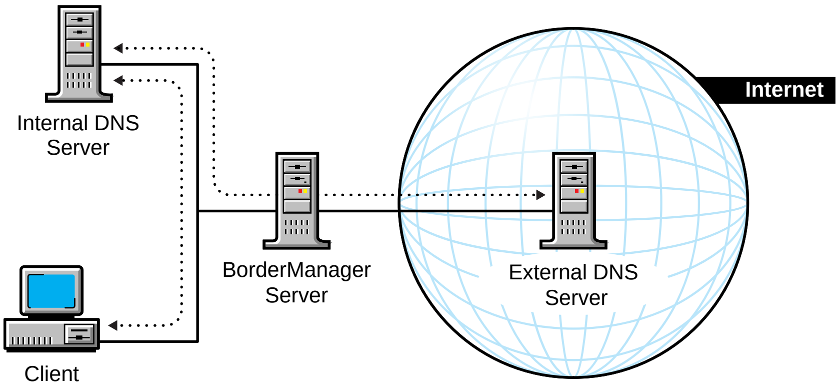


Figure 5-11 *Mail*



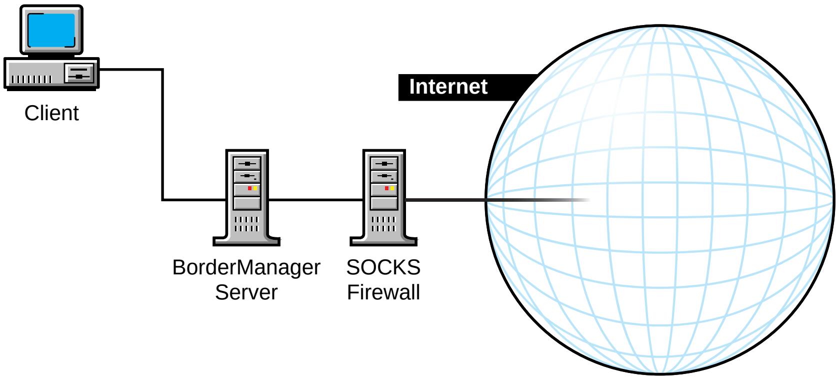
The Figure 5-12 on page 91 shows an example of using a DNS proxy. The Novell BorderManager proxy server configured for a DNS proxy handles traffic between the internal DNS name server and the DNS name server on the Internet.

Figure 5-12 *DNS Proxy*



The Figure 5-13 on page 91 shows an example of using a Novell BorderManager server behind an existing SOCKS firewall.

Figure 5-13 *Novell BorderManager Server behind a SOCKS Firewall*



Novell BorderManager Planning Scenarios

You can plan Novell® BorderManager® 3.9 Proxy Services by keeping in mind the following considerations:

- Section 6.1, "Adding an Inbound and Outbound Firewall," on page 93
- Section 6.2, "Adding an Outbound Firewall," on page 94
- Section 6.3, "Connecting an IPX-Only Site to the Internet," on page 95
- Section 6.4, "Configuring Multiple Virtual Private Networks," on page 96
- Section 6.5, "Adding Novell BorderManager to a Site That Already Has a Firewall," on page 98
- Section 6.6, "Using Novell BorderManager As an Address Translator," on page 99

6.1 Adding an Inbound and Outbound Firewall

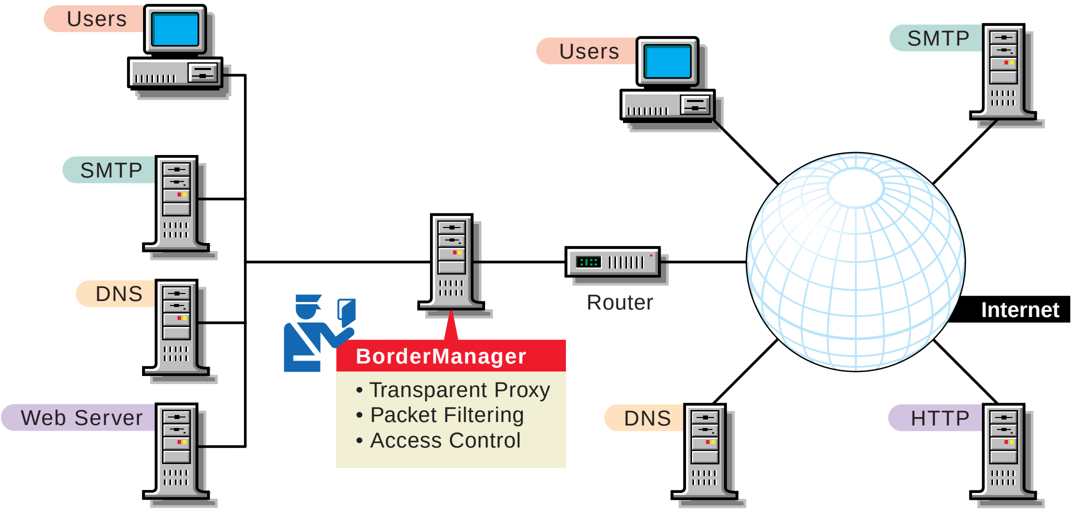
In this scenario, Company A is running TCP/IP and the Internetwork Packet Exchange™ (IPX™) protocol on the network. Company A wants to use Novell BorderManager as both an inbound and an outbound firewall. Company A wants to do the following:

- Add a firewall to secure the network
- Allow outbound and inbound Simple Mail Transfer Protocol (SMTP) e-mail
- Allow outbound and inbound DNS information
- Allow public users from the Internet to view only the Web server on the intranet
- Allow internal users on the intranet to access the Internet

The following Novell BorderManager components are used to implement this scenario, as shown in Figure 6-1 on page 94:

- Packet filtering
- Proxy Services Transparent HTTP proxy application
- Access control

Figure 6-1 *Inbound and Outbound Firewall*



To implement Novell BorderManager as a firewall on the network, Company A must perform the following general sequence of steps:

1. Install Novell BorderManager and enable packet filtering on public interfaces during the installation.

   For more information and for Novell BorderManager installation procedures, see *Novell BorderManager 3.9 Installation Guide*

2. Configure packet filtering.
3. Enable and configure the transparent proxy application on the Novell BorderManager server.
4. (Optional) Enable and configure the HTTP reverse, or acceleration, proxy to enhance performance.
5. (Optional) Enable and configure access control rules for the intranet users.

   For more information on configuring proxy services and access rules, see *Novell BorderManager 3.9 Administration Guide*.

6.2 Adding an Outbound Firewall

In this scenario, Company A is running TCP/IP on the network. Company A wants to use Novell BorderManager as an outbound firewall only, as shown in Figure 6-2 on page 95. Company A wants to be able to do the following:

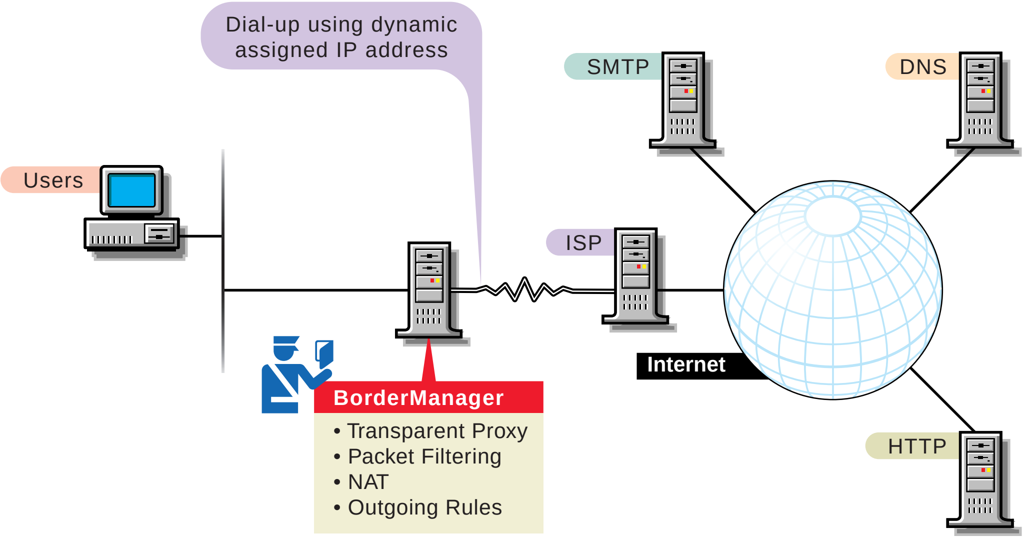
- Add an outbound-only firewall to secure the network
- Use a dial-up connection to the Internet Service Provider (ISP)
- Allow only internal users on the intranet to access the Internet
- Prevent Internet users from accessing or viewing the intranet

The following Novell BorderManager components are used to implement this scenario:

- Packet filtering
- Network Address Translation (NAT)

- Proxy Services Transparent proxy application
- Access control

Figure 6-2 Outbound-Only Firewall



To implement Novell BorderManager as an outbound-only firewall on the network, Company A must perform the following general sequence of steps:

1. Install Novell BorderManager and enable packet filtering on public interfaces during the installation.

   For more information on Novell BorderManager installation procedures, see *Novell BorderManager 3.9 Installation Guide*.
2. Use iManager enable and configure NAT for the WAN call to the ISP.
3. Use tiManager to enable and configure the Transparent proxy application.
4. Using iManager to enable and configure access control rules for the intranet users.

   For more information on configuration procedures, see *Novell BorderManager 3.9 Administration Guide*.

6.3 Connecting an IPX-Only Site to the Internet

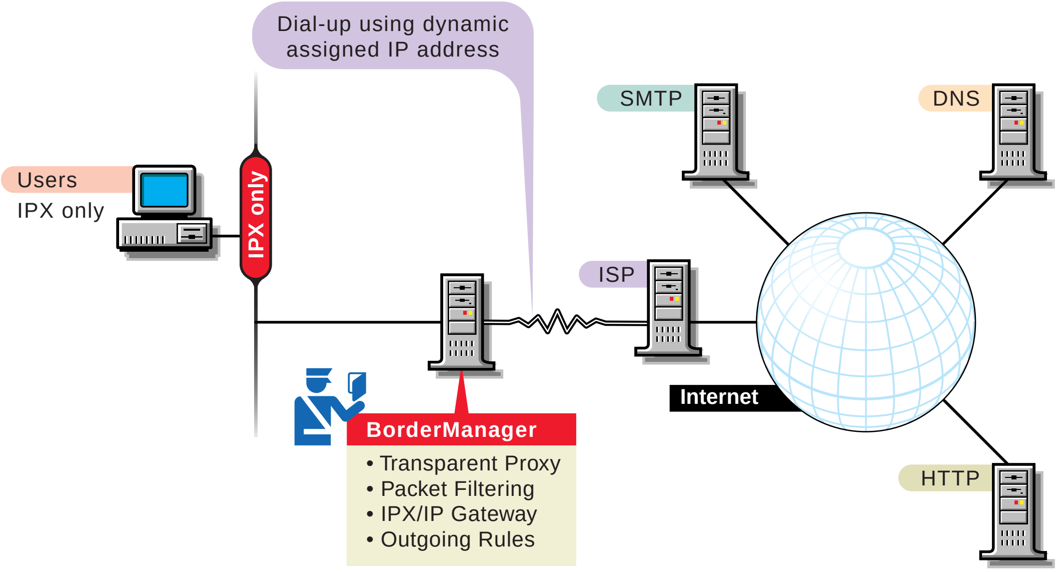
In this scenario, Company A is running only IPX™ on the network. Company wants to use Novell BorderManager to allow only intranet users to access and browse the Internet. Company A has the following requirements:

- Use only IPX and not require TCP/IP on workstations
- Allow only internal users to access the Internet
- Prevent Internet users from accessing or viewing the intranet
- Use dial on-demand to an ISP for the Internet connection

The following Novell BorderManager components are used to implement this scenario, as shown in Figure 6-3 on page 96:

- Packet filtering
- Proxy Services Transparent proxy application
- Access control

Figure 6-3 *Connecting an IPX-Only Site to the Internet*



To implement Novell BorderManager to connect to the Internet, Company A must perform the following general sequence of steps:

1. Set up the Novell BorderManager server to use dial on-demand routing to the ISP.
2. Install Novell BorderManager and enable the default packet filters and access control during the installation.

   For more information and for Novell BorderManager installation procedures, see *Novell BorderManager 3.9 Installation Guide*.
3. Using iManager, enable and configure the Transparent proxy application.
4. Using iManager, enable and configure access control rules for the intranet users.

   For more information and configuration procedures, see *Novell BorderManager 3.9 Administration Guide*.

6.4 Configuring Multiple Virtual Private Networks

In this scenario, Company A has remote users and two remote offices that must have a secure connection to the company intranet. Company A has the following requirements:

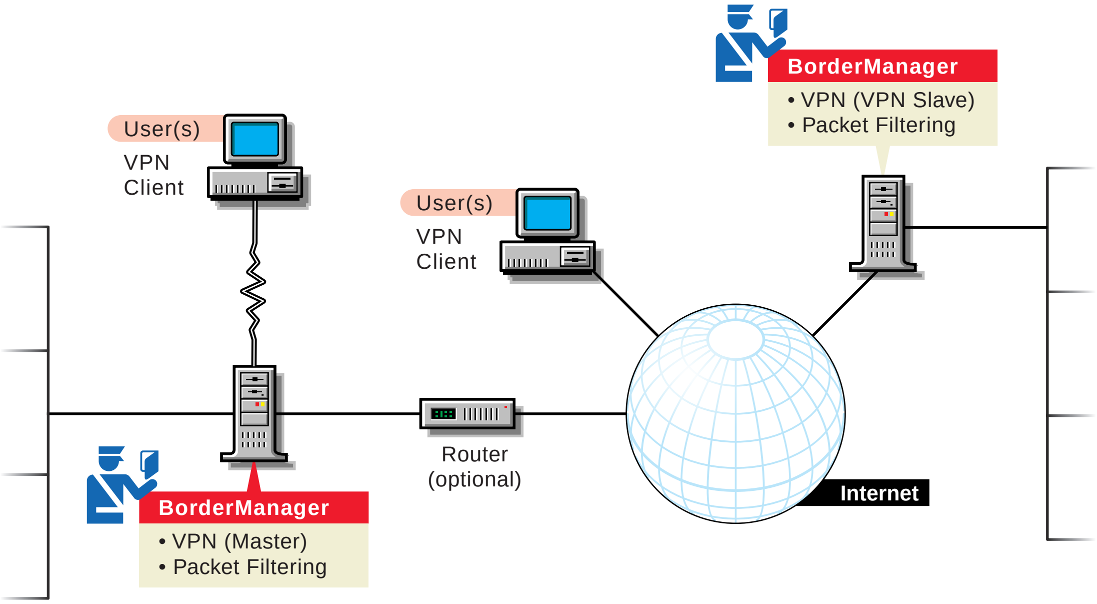
- Use server-to-server Virtual Private Networks (VPNs)
- Allow client VPN dialing directly into the VPN server
- Allow client VPN dialing into an ISP and then connecting to a master VPN server
- Provide the router with a permanent connection to the Internet

The following Novell BorderManager components are used to implement this scenario, as shown in Figure 6-4 on page 97:

- Packet filtering
- VPN server
- VPN client
- Access control

NOTE: In this scenario, on-demand links cannot be used, and a VPN server cannot be located behind NAT.

Figure 6-4 *Multiple VPNs*



To implement multiple VPNs, Company A must perform the following general sequence of steps:

1. Enable default packet filtering. This denies the default firewall filters, allowing VPN traffic while restricting other traffic.

   For more information and packet filtering configuration procedures, see *Novell BorderManager 3.9 Installation Guide*.
2. Install and configure the remote access software on the master VPN server.
3. Install and configure the master VPN server.
4. Install and configure the slave VPN server.

   For more information and configuration procedures, see *Novell BorderManager 3.9 Administration Guide*
5. Configure the VPN remote client.
6. Enable and configure access control rules allowing users to use the VPN client.

   For more information and configuration procedures, refer to Managing Access Control.

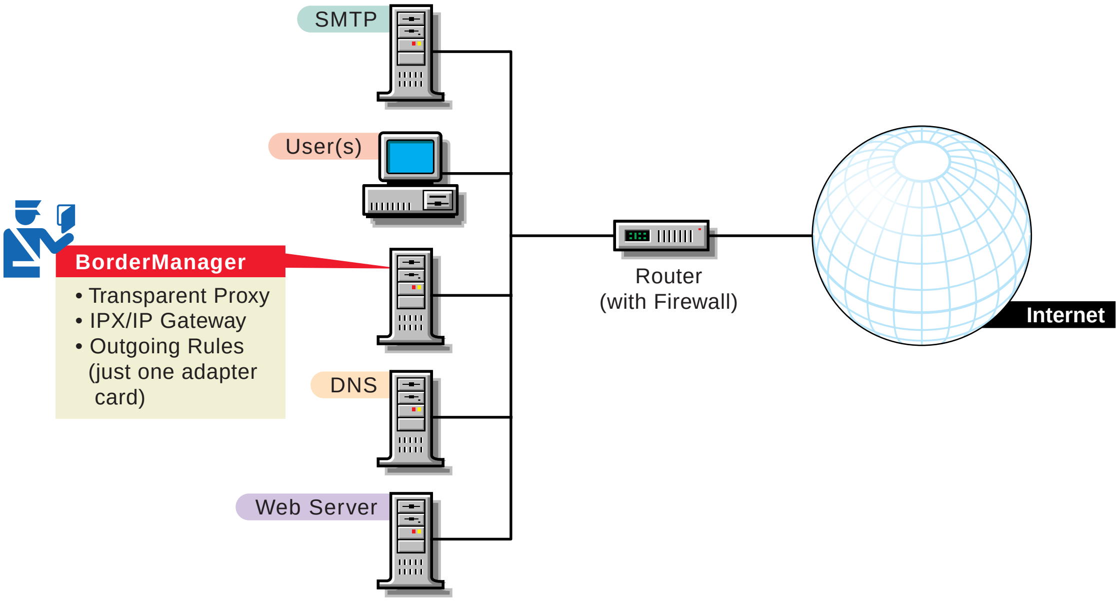
6.5 Adding Novell BorderManager to a Site That Already Has a Firewall

In this scenario, Company A already has a third-party firewall in place and wants to add Novell BorderManager as a proxy server. Company A wants to be able to just add the server onto the existing network. Company A has the following requirements:

The following Novell BorderManager components are used to implement this scenario, as shown in Figure 6-5 on page 98.

- Proxy Services Transparent proxy application
- Access control (optional with the existing firewall)

Figure 6-5 *Site that already has a Firewall*



To add a Novell BorderManager proxy server to an existing firewall, Company A must perform the following general sequence of steps:

1. Install Novell BorderManager. Because there is only one NIC, make sure it is selected as both Private and Public during the installation.

   For more information and for Novell BorderManager installation procedures, see *Novell BorderManager 3.9 Installation Guide*.

2. Enable and configure the Transparent proxy application on the Novell BorderManager server.
3. (Optional) Enable and configure access control rules.

   For more information on configuration, see *Novell BorderManager 3.9 Administration Guide*.

6.6 Using Novell BorderManager As an Address Translator

In this scenario, Company A wants all users on the private network to be able to access the Internet without registered IP addresses. Company A also wants to make the SMTP and Web servers on the intranet available to public clients. Company A has the following requirements:

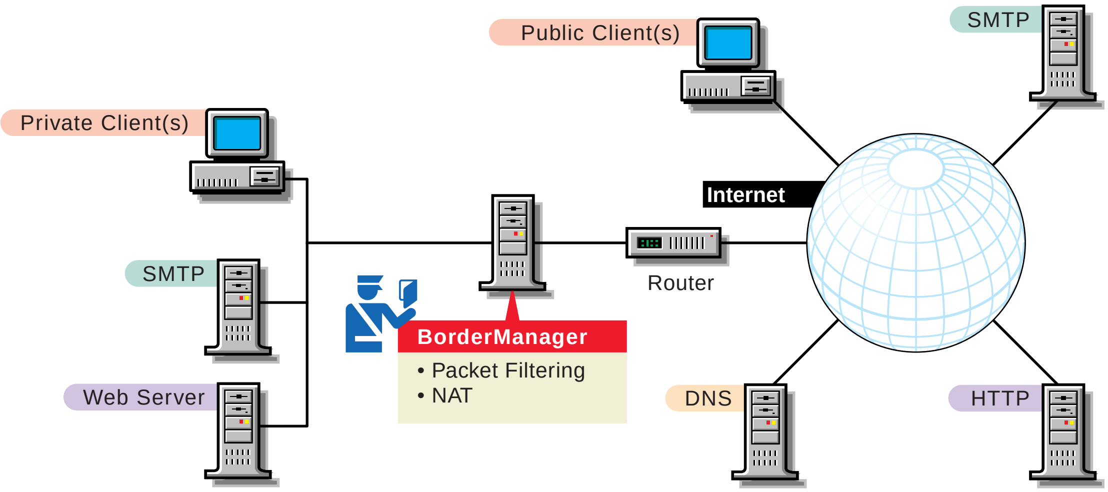
- Configure a Novell BorderManager server with NAT in dynamic mode to allow private users to access the Internet
- Make private SMTP and Web servers available through NAT static mode
- Consider filter settings for SMTP and Web servers on the NAT interface

The following Novell BorderManager components are used to implement this scenario, as shown in Figure 6-6 on page 99:

- Packet filtering
- NAT

NOTE: This scenario might not apply if your intranet Web server has links to other intranet Web servers, or if your intranet SMTP server has links to other intranet SMTP servers.

Figure 6-6 *Using Novell Novell BorderManager as an Address Translator*



To use Novell BorderManager as an address translator, Company A must perform the following general sequence of steps:

1. Install Novell BorderManager, enabling packet filtering on public interfaces.

   For more information and Novell BorderManager installation procedures, see *Novell BorderManager 3.9 Installation Guide*.

2. Enable the following filters:

   - For the intranet SMTP server, insert filter exceptions on the NAT interface to allow inbound SMTP requests and outbound SMTP responses.
   - For the intranet Web server, insert filter exceptions on the NAT interface to allow inbound HTTP requests and outbound HTTP responses.

3. Enable and configure NAT to use dynamic and static mode.

   For more information on configuration procedures, see *Novell BorderManager 3.9 Administration Guide*.