CHAPTER

# 6

# Redefining the DMZ: Securing Critical Systems

Yesterday's notion of the DMZ is dead! That's a bold statement, so let's back up just a moment and define the term "DMZ," which has taken a variety of meanings over the last decade. The original, basic definition of a DMZ as it relates to information security was:

> *"Short for demilitarized zone; a computer or small sub-network that sits between a trusted internal network, such as a corporate private LAN, and an untrusted external network, such as the public Internet."*—Webopedia, http://www.pcwebopaedia.com

By this definition, a DMZ refers to an area between a border router and a firewall, or *inside* the border router but *outside* the firewall, as you can see in Figure 6-1.

More recently, the following definition for DMZ has taken hold:

> *"A DMZ is a network or part of a network, separated from other systems by a firewall, which allows only certain types of network traffic to enter or leave. In a typical example, a company will protect its internal networks from the Internet with a firewall, but will have a separate DMZ to which the public can gain limited access. Public web servers might be placed in such a DMZ."*—Wikipedia, http://www.wikipedia.org

This definition is slightly broader, as you can see in Figure 6-2.

Is a DMZ still the area *outside* the firewall but *inside* the border router? Alternatively, is a DMZ a physical LAN "hanging off" of a firewall, or could a DMZ be *inside* the firewall? Does it really matter what you call a DMZ, or where it is physically located? Many variations of these definitions exist, but these examples will suffice to support our arguments that the

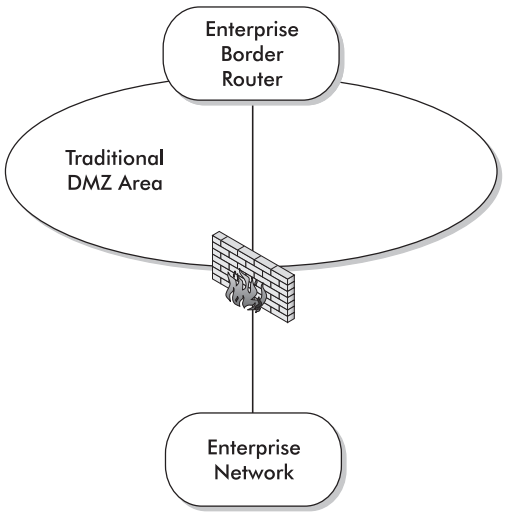


**Figure 6-1** *A traditional DMZ*

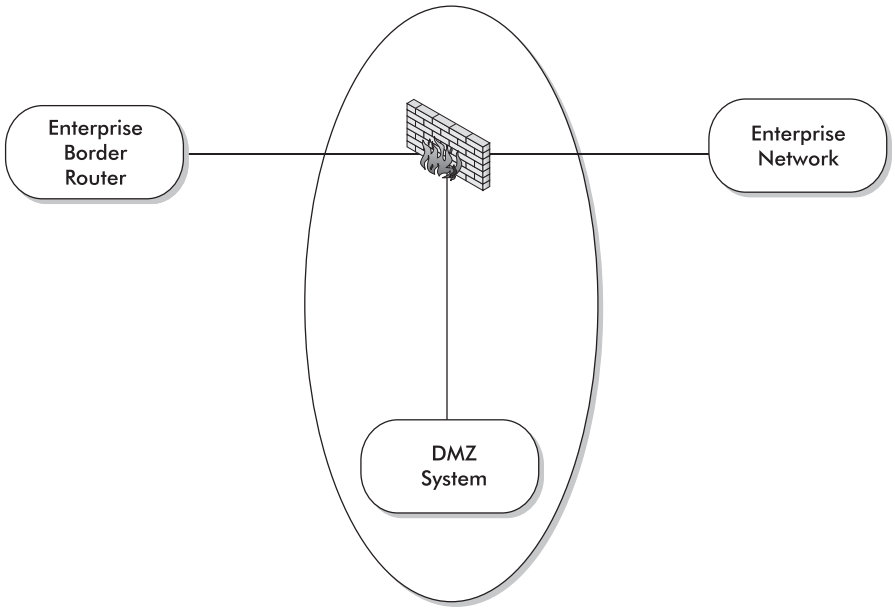


**Figure 6-2** *A modern DMZ*

notion of a traditional DMZ is dead. Instead, you should think of your network in terms of "security zones," each with its own security policies for access to systems and information within that zone.

The original idea behind a DMZ was to place the "sacrificial lamb" (system) in a non-trusted segment of your network. Today, a DMZ is typically an additional interface "behind" a firewall. If an attacker compromises the system, the *hope* is that internal trusted systems would be unaffected or unreachable by the attacker. If you hope that the physical DMZ will protect your internal trusted network, we would ask questions such as:

- Do you access DMZ systems from internal trusted systems?
- Do the DMZ systems use authentication directories that reside on the internal network?
- Do DMZ systems access backend databases inside the trusted network?
- Do you store proprietary or confidential information on the DMZ systems?

If the answer to any of these questions is "yes," then your hope may be unfounded.

Today's network must provide ubiquity, availability, and security of information. It's less about placing critical information assets (servers, databases) in a DMZ with special filters, and more about segregating access to those assets by *function*, wherever they reside throughout the entire network. Business is about secure, real-time access to specific data, from anywhere

in the world, 24 hours a day. Think about a typical business today and the functions and data that users might require access to, for example:

- Real-time electronic mail
- Customer Relationship Management databases (CRM)
- Web-based corporate applications
- Remote, secure access to the internal network
- Customer or supplier extranet

Now, are you going to deploy purpose-built mail gateways, customer database servers, and VPN concentrators "outside" your firewall, with replicated data, so that if these systems are compromised, your internal network is safe and secure? We think network security goes much deeper than that. Again, think of different parts of your network as security zones, and take a defense-in-depth approach to designing your network and systems infrastructure. You must identify the risks to your business, develop an all-encompassing security policy, and then implement layered security, consisting of physical, logical, and procedural security mechanisms. All of this is dependent on where a system is located in your network, what data resides on that system, which users need access to the system, and what they need to access. We are *not* saying that deploying specific systems inside some type of DMZ is inherently bad. We *are* saying that defense-in-depth extends beyond the concept of a DMZ, from your border router all the way in to your internal network.

This chapter will discuss components, methods, weaknesses, and a checklist for securing critical systems.

- **Components of Defense-in-Depth** A brief explanation of a defense-in-depth strategy.
- **Exposing Weaknesses of DMZs** How attackers can impact security through weaknesses in various DMZ implementations.
- **Stand-alone Systems in the DMZ** Discussion of strengths and weaknesses of replicated data on a stand-alone system in the DMZ.
- **Reverse-Proxy Systems in the DMZ** Discussion of strengths and weaknesses of a hardened, reverse-proxy system in the DMZ, communicating securely with backend systems in the internal network.

# Components of Defense-in-Depth

As stated in the introduction to this chapter, the concept of a DMZ is just too limited with respect to securing critical systems. Simply deploying hardened systems in a traditional DMZ outside the firewall doesn't give you much depth in security. In this chapter, we cover the following components related to securing any type of DMZ system, building on what we covered in earlier chapters:

- Defining "security zones" throughout your entire network infrastructure
- Allowing strict border router access control lists (ACLs)

- Hardening the operating system of DMZ hosts
- Providing secure authentication and authorization mechanisms to access DMZ systems
- Ensuring strict trust relationships between DMZ systems and both internal and external systems
- Extending specific segments of your network through VPN mechanisms

## Defining Security Zones

What is a "security zone"? If you think of the DMZ as a moat between two perimeter walls, this may be one of your zones. Additionally, your border router may be considered another zone, and your firewall may be considered a zone. When taken as a whole, the area from your border router, through the DMZ and firewall, to an internal system may be another zone. We can't define security zones for you, because defining a zone is dependent upon each organization's networks, systems, and security policies. However, you should think in terms of the end-to-end "zone" that encompasses securing critical systems, as well as smaller zones that encompass the whole. For example, if you deploy a system in a traditional DMZ (on the LAN between a border router and firewall), the security zone for that system might consist of the following components:

- Border router access control lists
- Layer 2 infrastructure connecting border router, DMZ, and firewall
- DMZ system (hardening the operating system, securing local data, securing authorization and authentication)
- Firewall access control lists, permitting/denying specific protocols/ports between the DMZ system and internal systems (trust relationships)

These zones are depicted by the shaded area of the network diagram in Figure 6-3. In addition, the firewall may be a subzone since it forms the perimeter between your internal network and all external networks.

Alternatively, let us assume that the DMZ system has localized authentication/authorization data and a database accessed by customers. The system never contacts backend systems within your internal network, and you manage the system from the console only. The security zone is now smaller, as depicted in Figure 6-4.

We could give many more examples, but they may or may not be relevant to your particular infrastructure. We simply want you to start thinking about all network "pieces" associated with securing a critical system. Then define your security zone once you have identified the following components related to the system you're securing from end to end:

- All network elements
- All protocols and services accessed
- External access policies (what is allowed between the Internet and DMZ system)

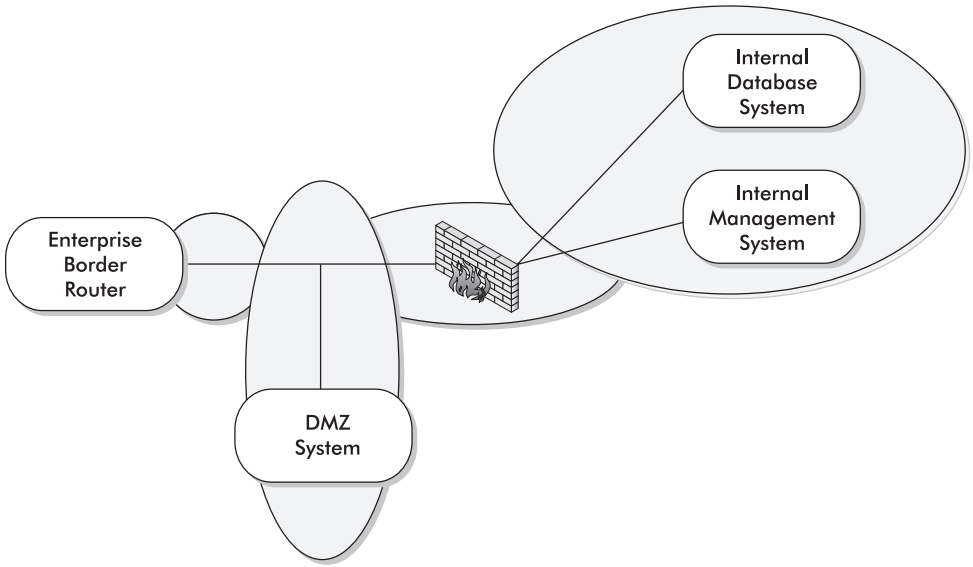


**Figure 6-3** *An example of an end-to-end DMZ security zone*

- Internal access policies
  - Will the DMZ system be managed from internal systems?
  - Will the DMZ system access data stores on the internal network?
  - Who needs access, when do they need access, and what level of access do they need (read-only, modify, super-user)?

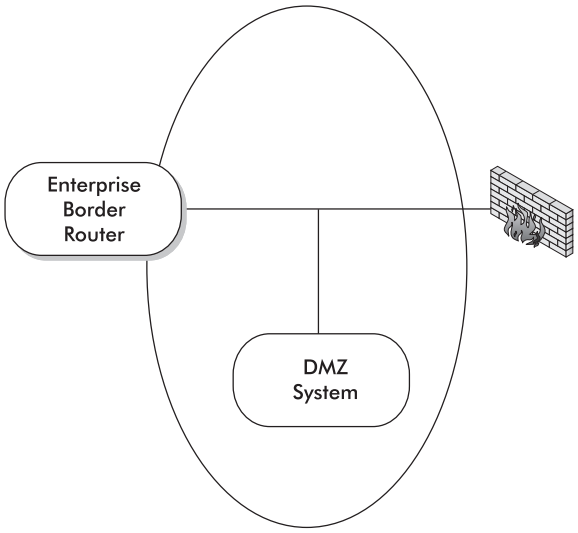


**Figure 6-4** *An alternative end-to-end DMZ security zone*

# Exposing Weaknesses of DMZs

Table 6-1 lists some common weaknesses in DMZ design and how attackers may exploit them to disclose proprietary/confidential information, or penetrate further into your network.

# Stand-alone Systems in the DMZ

As we noted in the introduction, a traditional DMZ typically consists of stand-alone systems residing between a border router and a firewall, while a modern DMZ typically consists of stand-alone systems behind a firewall, yet not on the *internal* network. A more advanced

| Potential Weakness in DMZ Design | How the Weakness May Be Exploited |
|---|---|
| Insufficient ingress filtering on border router. | Attackers may find a hole in ingress filters giving unintended access to services on the DMZ system or giving access to the border router. |
| Insufficient hardening of DMZ systems. | You may have strict ingress and/or firewall filtering, but attackers find a weakness in the operating system or services on the DMZ system. |
| Open trust relationships between DMZ systems and other internal/external systems. | Attackers may exploit weaknesses in trust relationships between DMZ systems and backend database servers or authentication servers, resulting in information disclosure or further penetration into your network. |
| Replicated data resides locally on the DMZ system. | If attacker compromises DMZ system, you may inadvertently disclose proprietary/confidential corporate or customer information. |
| User authentication data resides locally on the DMZ system. | If authentication data is replicated from internal systems, or exists on other DMZ systems, attackers that compromise one system may be able to access other systems as an authorized user. |
| Lack of event logging from border routers, DMZ systems, Intrusion Detection Systems, or firewalls. | Any part of the network infrastructure may be compromised, and without proper event logging, you may never know! |

**Table 6-1** *Potential Weaknesses in DMZ Design and Methods of Exploitation*

DMZ design consists of hierarchical firewalls. For organizations utilizing these methods, we point out potential weaknesses and methods to mitigate risk of intrusion.

We see three basic designs in deployment of DMZ systems:

- Traditional DMZ
  - Border router (screening router)
  - DMZ system(s)
  - Firewall (with internal network behind the firewall)
- Modern DMZ
  - Border router (screening router)
  - Firewall
    - External interface
    - Internal interface
    - One or more "DMZ" interfaces on the firewall
- Advanced Hierarchical Firewalls
  - Border router (screening router)
  - Perimeter firewall
  - DMZ inside perimeter firewall
    - DMZ systems reside here
  - Internal firewall

## Traditional DMZ

Again, a traditional DMZ may be viewed as a "moat" separating the outside wall from the inside wall. While not as common today, we still see this type of DMZ deployed. We don't advocate this type of design but it *can* be made secure with proper use of security zones. A myopic view would see this as the DMZ security zone. We posit that the zone *includes* the border router (access control between the Internet and the DMZ) and the firewall (access control between the DMZ and the internal network). Figure 6-5 depicts a traditional DMZ.

This design is simple and effective if secured properly. Building on what you learned in Chapter 4, you should define the security zone for systems in a traditional DMZ, which should include the border router, the firewall, and the DMZ system. The checklist in Table 6-2 may be used to develop defenses throughout the zone.

There are benefits in using a properly secured, traditional DMZ design, but this method doesn't scale well in large networks. Table 6-3 summarizes some of the risks and benefits of this approach.

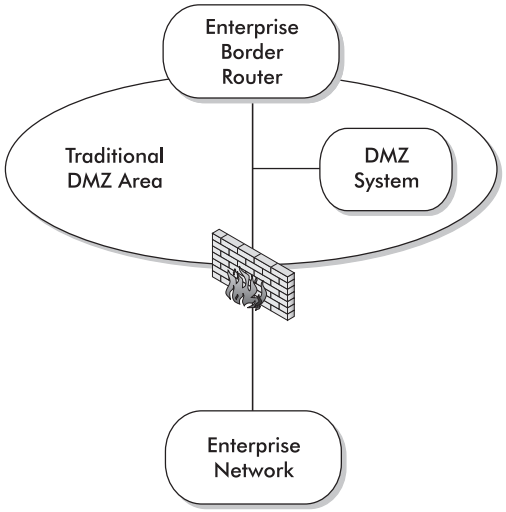


**Figure 6-5** *A traditional design for a DMZ*

| Security Zone Component | Defensive Technique |
|---|---|
| Border router access control | Explicitly allow only those protocols/services necessary for Internet users to access the system; deny everything else |
| DMZ host operating system | Harden the host operating system, disable all unused/dangerous services, and ensure patches are current |
| System administration | Use a secure remote administration mechanism (encryption), or console-only access. Use strong authentication and authorization methods, and data integrity functions (TripWire, etc.) |
| Trust relationships between DMZ host and internal/external systems | Where possible, use VLANs to restrict traffic between DMZ hosts (if more than one), and/or use host-based firewall to establish trust relationships between other systems, perform egress filtering on both border router and firewall |
| Firewall access control | Ensure strict access control for trusted internal systems to access DMZ host, and for DMZ host to access trusted internal data stores (authentication data, application data) |
| Intrusion Detection/Prevention Systems (IDS/IPS) | Proper logging and IDS/IPS can alert you to possible attacks before they succeed |
| Flow statistics collection on the border router | Correlate flow data with IDS/IPS and system logging to provide rapid alert mechanism in case of attack/intrusion |

**Table 6-2** *Checklist for Developing Defenses in a Traditional DMZ Security Zone*

| Risks | Benefits |
|---|---|
| Capital expense may be prohibitive if you have hundreds or thousands of systems to deploy. | If systems are completely isolated from the internal network, compromise is limited to that specific system (complete isolation is rare occurrence; there is typically some trust relationship with internal systems). |
| If proprietary/confidential data is stored locally on DMZ system, data theft/disclosure is more likely. | Stand-alone systems can be "lean and mean," optimized to serve a specific application. |
| Weak trust relationships/filtering with internal systems may give attackers the "keys to the kingdom" (this applies to all designs of DMZ networks). | Reduced resource utilization on the firewall since DMZ hosts are external. |
| A border router cannot typically provide in-depth, stateful inspection of packets destined for the DMZ host, and may leave it vulnerable to specific attacks. | |

**Table 6-3** *Risks and Benefits of a Traditional DMZ Design*

# Modern DMZ

A more common DMZ design we see today is a DMZ LAN attached to a tertiary interface on a firewall. One or more interfaces may be used for DMZ systems, in addition to the untrusted/external and the trusted/internal interface. In this case, the security zone might include the border router, external network (between the border and firewall), the firewall, the DMZ LAN, and possibly the internal LAN. As you can see in Figure 6-6, this zone covers many security devices.

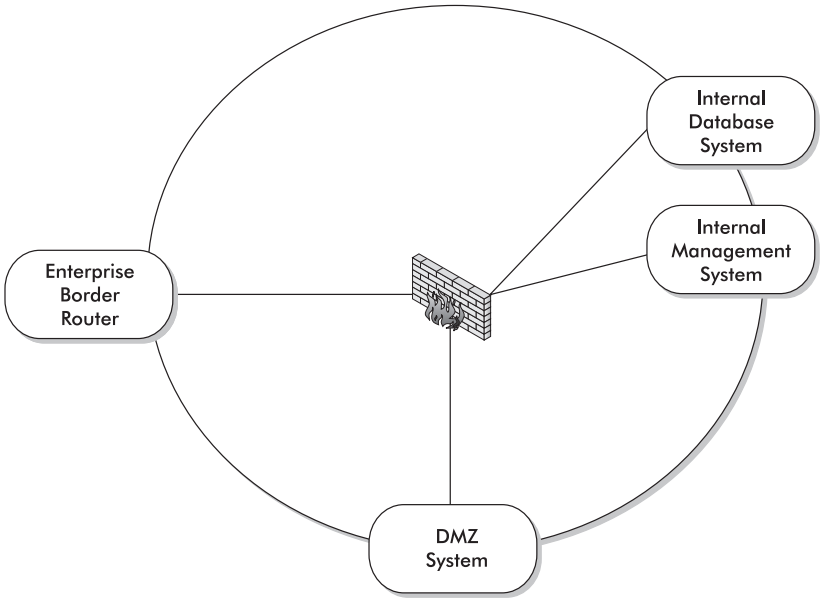


**Figure 6-6** *A modern DMZ design*

| Risks | Benefits |
|---|---|
| Higher resource utilization on the firewall, since it is now protecting the host. | Combination of border router and firewall filtering provides greater defense-in-depth. |
| If DMZ host is compromised, attacker may be "deeper" inside your infrastructure. | Stateful packet inspection and strict filtering on the firewall provides more granular protection for DMZ host. |
| Capital expense may be prohibitive if you have hundreds or thousands of systems to deploy. | More granular control between DMZ and internal network, if DMZ host accesses data stores there. |

**Table 6-4** *Risks and Benefits of a Modern DMZ Design*

This design is seen more frequently today, and many variations of this design exist. The checklist for developing defenses in Table 6-2 applies to this design also. However, this design provides more granular control and flexibility than the traditional DMZ. Table 6-4 summarizes some of the risks and benefits of this approach.

## Advanced DMZ Design Using Hierarchical Firewalls

A more advanced design for modern DMZs may consist of two or more firewalls layered within the network topology to provide maximum benefit of stateful packet filtering and access control between different security zones. An example of this design is depicted in Figure 6-7.

For small networks, this design may be overkill, but in very large networks, this design can provide a high degree of flexibility, segregation of security zones, and trust between zones. The entire network in this example may be considered a security zone, but it makes more sense to segregate this network into at least three distinct zones; we'll call them the

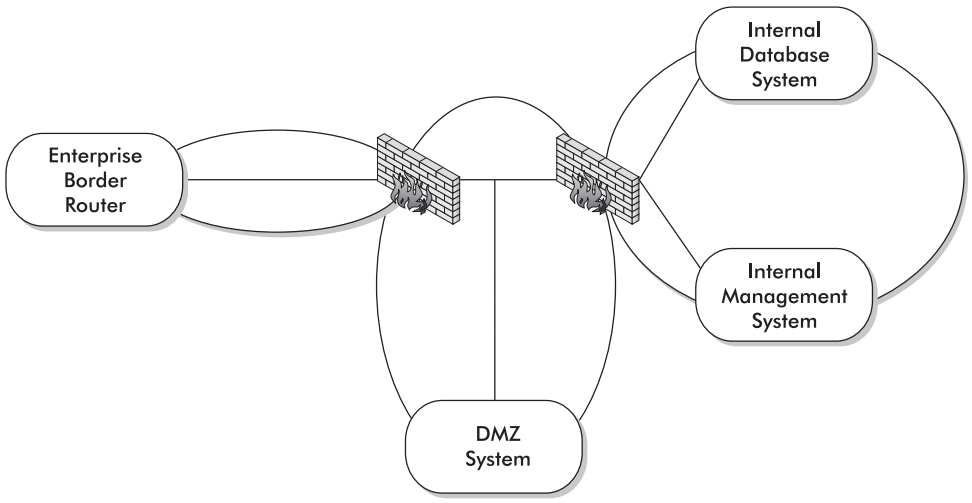


**Figure 6-7** *A DMZ design with hierarchical firewalls*

Perimeter, DMZ, and Internal zones. Given this segregation, you may wish to deploy systems in all three zones and establish specific trust relationships and packet filtering based on the function of these systems. The following table provides an example of systems you might deploy in each zone and the relationships between those systems.

| Security Zone | System Deployed | Trust Relationships |
|---|---|---|
| Perimeter | Mail relay | Corporate mail server and management console in Internal zone |
| Perimeter | Corporate Web Server (Internet information site) | Management console in Internal zone |
| DMZ | Extranet Web Server (customer web portal) | Backend database server and management console in Internal zone |
| Internal | Management Console | Systems in DMZ and Perimeter zones |

This is a complex design, with nuances too numerous to cover in this chapter, but the following table enumerates some of the risks and benefits of this design.

| Risks | Benefits |
|---|---|
| Policy management is more complex, and human error may lead to unintended attack or intrusion. | More granular filtering and policy enforcement between security zones. |
| Capital and operating expense may be prohibitive due to additional systems to manage and complexity of policy enforcement. | Strict policy enforcement between zones helps isolate intrusion to a specific zone. |
| | Denial-of-service attacks are more difficult to sustain. |

# Reverse-Proxy Systems in the DMZ

One of the most cost-effective and secure methods of providing access to critical systems is through a reverse-proxy. *Proxy* simply means to "act on behalf of another entity." People typically think of proxy as a *forward-proxy*, which is done through firewalls using NAT, web caches, or web proxies, from internal users to the Internet. A proxy typically has a *many-to-one* relationship, such as a firewall with NAT, mapping many internal users with private addresses to a single globally routed address outside the firewall. An example of a simple many-to-one web-based proxy is shown in Figure 6-8. The proxy may be used to provide a centralized mechanism for web content filtering, web content caching, or user authentication.

The concept of a *reverse-proxy* is the same as a forward-proxy, but think of traffic flowing in the reverse direction (Figure 6-9), from the Internet in to secure systems in your network.

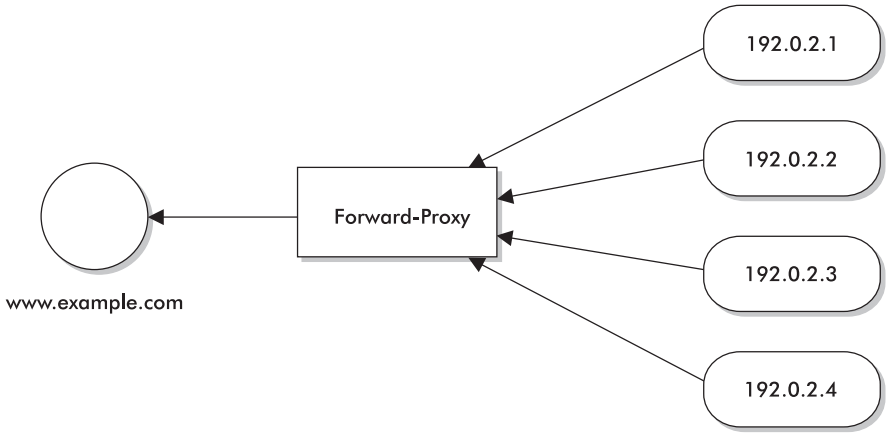


**Figure 6-8** *A simple example of a forward-proxy*

Let us assume that you wish to provide the following applications to your corporate users from any location on the Internet, through a reverse-proxy:

- Electronic mail
- Corporate file shares
- Corporate address book and/or calendars
- Customer Relationship Management (CRM) database

You can probably imagine that each of these applications may provide access to confidential and/or proprietary information, and each may run on a different platform. In addition, each application may have different authentication mechanisms. However, one thing each of these applications *probably* has in common is a web-based client access mechanism.

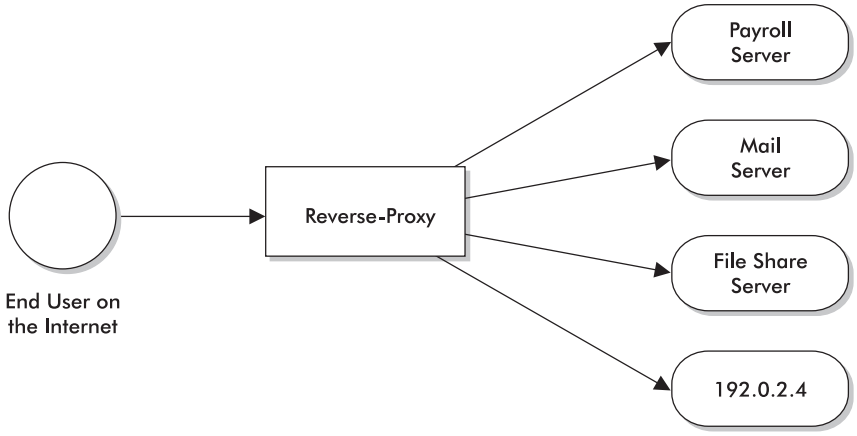


**Figure 6-9** *A simple example of a reverse-proxy*

Given all of the DMZ designs we described previously, which one appears best suited for deployment of these applications? All but the hierarchical firewall design seem to have risks that obviate their use for these applications. However, with the use of a reverse-proxy system, even the traditional DMZ deployment can be made secure for these applications.

***NOTE***

*We are assuming that each application has a web-based client access mechanism. However, many applications today will function through a web-based reverse-proxy, even if they do not contain web-based, client access middleware such as Java.*

There are a variety of examples we could present for a reverse-proxy in the DMZ. Instead, we present a real case study from a current customer of ours.

## Reverse-Proxy Case Study

Problem: A small company of approximately 250 employees repairs and maintains heavy construction equipment. The company has field service trucks that travel to remote construction and mining sites to repair heavy equipment. The drivers rarely visit the corporate offices, so they need secure Internet access to check parts inventory at the corporate office, send and receive e-mail, access corporate file shares to view operator manuals, enter work-time in the time-keeping database, and obtain anti-virus updates. The work sites are typically so remote that even cell phone access is unavailable, so the company implemented a satellite communication system. The satellite network provider interconnected with Internet service providers, which provided the field trucks with Internet access. The company attempted to use IPSec VPN access over the satellite system, but the delay imposed by the long round-trip of packets over satellite rendered IPSec unusable.

The company had the following requirements for its system:

- It must perform well over a satellite communication network.
- All communication between field trucks and the corporate network must be encrypted.
- It must support Microsoft Outlook Web Access (OWA).
- It must support Microsoft file sharing.
- It must support Microsoft NTLM authentication and cached credentials.
- It must support anti-virus signature definition updates.
- It must support terminal emulation for an interactive, command-line parts inventory database on an IBM AIX system.

Solution: We proposed a reverse-proxy system deployed behind the corporate firewall. The reverse-proxy was a hardened Linux system running an Apache web server with SSL and digital certificate-based authentication. The Apache server also included a module for pass-through NTLM authentication to Active Directory. Users would connect to the

reverse-proxy, authenticate, then the reverse-proxy presented users with a menu of internal systems to access, including OWA, file shares (through WebDAV), the time-keeping system, and the parts inventory system.

While NTLM authentication functioned transparently with the time-keeping system and the parts inventory system, the pass-through authentication did not function with OWA or WebDAV file shares, since these applications already utilize NTLM authentication. In this case, the reverse-proxy did *not* authenticate the user, but simply passed the user's request to the appropriate system, and authentication was performed from the end system to Active Directory. Terminal emulation for the parts inventory database was accomplished through a vendor-supplied Java client, which was "tunneled" through the SSL connection to the end users.

Given some of the budget and technical requirements of the customer, we had to customize some aspects of this design. The following caveats applied to this design:

- Microsoft Internet Explorer is the only browser supported, due to the requirement for NTLM authentication and cached credentials.
- The reverse-proxy was placed *inside* the firewall because the time-keeping system did not support HTTPS connections. The end-user connection *to* the proxy was encrypted with SSL (over the Internet), but the reverse-proxy communication was unencrypted to the time-keeping system.
- The browsers had to be configured with Integrated Windows Authentication to use NTLM with the reverse-proxy. In addition, browsers had to have trusted security zones enabled, with host names for all internal systems to be accessed.
- Since the reverse-proxy is behind the firewall and only a single globally routed IP address is "visible," the remote workstations had to be configured with specific host names for each internal server (application), which mapped to the single IP address.
- The reverse-proxy was configured with the <VirtualHost> tag, which maps the client's requested host to the *internal* IP address of the server behind the proxy.

This is just one example of a design for a reverse-proxy system but it fit the specific requirements perfectly for this customer. The table below enumerates some of the risks and benefits.

| Risks | Benefits |
|---|---|
| Policy management is more complex, and human error may lead to unintended attack or intrusion. | More granular filtering and policy enforcement between security zones. |
| Capital and operating expense may be prohibitive due to additional systems to manage and complexity of policy enforcement. | Strict policy enforcement between zones helps isolate intrusion to a specific zone. |
| | Denial-of-service attacks are more difficult to sustain. |

# A Checklist for Developing Defenses

| Step | Description |
|---|---|
| Develop security zones. | Identify all network elements that a critical system is dependent upon when defining a security zone. |
| Consider potential weaknesses in DMZ design. | Review Table 6-1 when designing a DMZ (or security zone). |
| Utilize reverse-proxy systems. | Reverse-proxies are flexible, scalable, economical, and can provide higher security than various DMZ designs when implemented properly. |

## Recommended Reading

- SANS Security Policy Project (http://www.sans.org/resources/policies/)
- Information Assurance Technical Framework (http://www.iatf.net/)
- Cisco SAFE: Security Blueprint for Enterprise Networks (http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safe_wp.pdf)
- NTLM Authentication (http://davenport.sourceforge.net/ntlm.html)
- NTLM Authentication with HTTP (http://www.innovation.ch/java/ntlm.html)