Module 17

E-Commerce Security Needs

CRITICAL SKILLS

17.1	Understand E-Commerce Services
17.2	Understand the Importance of Availability
17.3	Implement Client-Side Security
17.4	Implement Server-Side Security
17.5	Implement Application Security
17.6	Implement Database Server Security
17.7	Develop an E-Commerce Architecture

Electronic commerce, or *e-commerce*, has become a buzzword of the Internet. Organizations all over the world have appeared on the Internet to offer everything imaginable. Some of these endeavors have succeeded and some have failed spectacularly. One thing that the successful organizations have in common is the fact that they understand that they are doing e-commerce to make money. They may make money by providing a new service via the Internet, by expanding the reach of an existing service, or by providing an existing service at a lower cost.

Organizations who choose to perform e-commerce are taking a risk. They are investing in new technologies and new ways of providing goods and services in the hope of making a profit from the activity. The risks to the organization come from several areas: the public may not accept the service, the new customers may not appear, or existing customers may not like the new service. Because these organizations are performing e-commerce, a whole new set of threats and vulnerabilities must be taken into account. These new threats and vulnerabilities create new risks that must be managed.

One thing to keep in mind as we talk about e-commerce is that electronic ordering and payment systems have existed for a long time. Electronic Data Interchange (EDI) has been used between businesses to order goods and make payment for years. The big development that makes e-commerce a hot topic is that now regular consumers can order just about anything they want from whomever they want, and any organization can open a store within days of choosing to do so. In addition, many organizations that sold goods via large distribution channels can now sell directly to consumers and thus decrease their overhead costs.

CRITICAL SKILL 17.1

Understand E-Commerce Services

What kinds of services can e-commerce offer us? The list is long and some of the services are truly new and innovative. For example, some organizations are selling subscriptions to information. This type of service has been available in the past, but it was always expensive and it usually required a special dial-in line. Now anyone can access these services over the Internet. The service provider can also increase revenue by providing information to consumers at a lower cost.

Another service that has come with the advent of e-commerce over the Internet is providing electronic library functions for sensitive or confidential information. Organizations can subscribe to a service that stores and makes available their own information electronically. Delivery of the information back to the organization is via the Internet. For example, Organization A contracts with Vendor V to maintain and archive electronic information. Vendor V creates a data center with a large amount of storage and takes delivery of Organization A's files. These files are then placed on systems so that employees of Organization A can access them securely. Vendor V charges a fee to Organization A for the amount of data to be stored.

Other services that are provided through electronic commerce include functions that organizations have performed in the past but that may now be performed cheaper. A good example of this is distribution of information. Manufacturers, for example, need to distribute product information and price lists to networks of distributors or resellers. In the past, the

manufacturers have printed and sent the information in hard copy through the mail, or they set up elaborate and expensive private networks to allow the distributors to connect to the manufacturer and get the information. With e-commerce, the manufacturer can establish a single site on the Internet and allow the distributors and resellers to connect via the Internet and get the information they need. The service is both cheaper and timelier.

Probably the e-commerce service most commonly thought of is the purchasing of goods. Even here in a very traditional service, we can see innovation. Some organizations have taken to selling electronic books or music via MP3 files. The traditional service of selling goods is here as well. Many sites on the Internet provide the consumer with the ability to purchase goods. Consumers make an order and then the goods are sent to the consumer.

Differences Between E-Commerce Services and Regular DMZ Services

It is obvious that e-commerce services can be provided using similar infrastructures as those needed for Internet connectivity. Web servers, mail servers, and communication lines are all necessary. But there are differences between how e-commerce services are designed and how normal Internet services are designed.

The differences between the two begin with the requirements of the services. For regular Internet or DMZ services (see Module 16 for more information on DMZ), the organization wants to provide information to the public (Web sites) or transmit information between the organization's employees and the public (mail). The organization may want to verify that it is providing correct information over its Web site and that the Web site is usually up. The same is true for mail. The mail service is store and forward. Sometimes it takes a while for a message to be delivered. If inbound mail is delayed due to a system failure, it is not a big deal to the organization. Inbound mail is not critical for day-to-day business and thus the source of the e-mail does not need to be verified beyond the source e-mail address.

Now think about the requirements for commerce. The organization still wants to provide a service to the public (for business-to-consumer e-commerce, anyway); however, the organization must know who is ordering goods and who is paying for them. At the very least, the organization must verify the identity of the person ordering the goods. Since we do not have universal identity cards, the organization must use some other form of identification. Most often it is a credit card in conjunction with the shipping address for the goods.

Another new aspect of e-commerce services is the need to keep some information confidential. The information may be what is being sold (so that the organization is properly compensated for the information), customer information that has been held for safekeeping, or it may be the information used in the purchase (such as credit card numbers).

These two primary differences, verification and confidentiality, differentiate the e-commerce services from regular DMZ services. There is one other issue that must be taken into account when e-commerce is discussed. That is availability. No longer is the Web site just

for information about an organization. Now the e-commerce site generates revenue and provides a service to the customers. Availability becomes a critical security issue for the e-commerce site.

Examples of E-Commerce Services

When we think about applying security to e-commerce services, we can think in terms of the four basic security services discussed in Module 4: confidentiality, integrity, availability, and accountability. We can also assume that availability is an issue for any kind of e-commerce. The issues surrounding the other three services differ depending on the type of e-commerce service that you offer. The following sections provide three examples of how security may be needed around e-commerce services.

Selling Goods

Your organization wants to sell goods to the public via the Internet. The basic concept is that the public will come to your Web site, examine your goods, and order the goods for shipment. Payment will be provided through a credit card and the goods will be shipped using the most economical method.

Based on this scenario, we can examine the security requirements for each of the base security services:

- **Confidentiality** Most of the information is not confidential. However, the credit card number certainly is. The customer's e-mail address and other personal information may be as well, depending on the privacy policy of the site.
- **Integrity** The customer will want to have integrity in the order so that she gets what she wants. To keep the organization's books correct, we will need to guarantee the integrity of the order throughout the process. We will also need to guarantee the integrity of the catalog so that the price in the catalog is the price that is paid for the item.
- **Accountability** The organization will need to make sure that the person using the credit card is the owner of the card.

As you can see from this brief example, security will play a large role in the architecture of this e-commerce system.

Providing Confidential Information

Let's take a look at a different e-commerce service. In this example, the organization provides information to the public for a fee. The information that is provided is owned by the organization, and they will want to control how this information is shared. The organization sells access to the information to individuals or to other organizations on a subscription basis.

Based on this scenario, we can examine the security requirements for each of the base security services:

- **Confidentiality** All of the information provided to the customers is confidential and must be protected in transmission as well as after the customer gets the information. Payment is normally made through another mechanism (for the subscription service), so no credit card information must be handled by the e-commerce service.
- **Integrity** The customer will want to have integrity of the information provided, so there must be some assurance that information in the organization's database has not be tampered with.
- **Accountability** Since the customers purchase subscriptions to the information, the organization will need to have some form of identification and authentication so that only subscribers can view the information. If some customers are billed by their usage of the system, an audit trail must be kept so that billing information can be captured.

Distribution of Information

As a last example, let's take a manufacturing organization that uses distributors to sell its goods. Each distributor requires pricing information as well as technical specifications on current models. The pricing information may be different for each distributor, and the manufacturer considers the pricing information to be confidential. Distributors can also make orders for goods through the service and report defects or problems with products. Distributors can also check to see the status of orders previously made.

Based on this scenario, we can examine the security requirements for each of the base security services:

- **Confidentiality** Price sheets, orders, and defect reports are confidential. In addition, each distributor must be limited in which price sheets and orders can be seen.
- **Integrity** The price sheets must be protected from unauthorized modification. Each order must be correct all through the system.
- **Accountability** The manufacturer will need to know which distributor is requesting a price sheet or making an order so that the correct information may be provided.

CRITICAL SKILL 17.2

Understand the Importance of Availability

I am breaking out availability as a separate issue because it is the key issue for e-commerce services. If the site is not available, there will be no business. The issue goes deeper than this as well because the availability of the site impacts directly on the confidence a customer will have in using the service. Now this is not to say that failures in other security services will not impact customer confidence (you can just see recent failures in confidentiality to see the impact they have), but a failure in availability is almost guaranteed to push a potential customer to a competitor.

Business-to-Consumer Issues

We start our examination of availability with the issues associated with an organization that wants to do business with the general public or consumers. There are several issues surrounding availability. First, when does the consumer want to use the service? The answer is, whenever they want to use it. It does not matter when the organization thinks they will have customers, it only matters when the customers want to visit the site and do business. This means the site must be up all the time.

Also keep in mind that this means the entire site must be up all the time. Not only must the Web site be up, but also the payment processing and any other part of the site that a customer may want to use. Just think how a potential customer might feel if they find the site and identify the item they want to purchase only to find that the order cannot be processed because the payment system is not available. That customer is likely to go somewhere else.

While it is not a security issue, the whole problem of availability includes business issues such as the ability of the organization to fulfill the orders that are entered into the system. When building the site, the infrastructure should be sized for the expected load. There is a television commercial that illustrates this point very well. The commercial starts with a team of people who have just completed an e-commerce site. They are watching a screen and waiting for the first order. It appears, and everyone breathes a sigh of relief. Then more orders come and more and more until the scene closes with several hundred thousand orders. It is obvious from the reactions of the team that they were not expecting this and they may not be able to handle it. Such issues hit online retailers over the 1999 Christmas season. Several large retailers had trouble handling the number of orders and almost went out of business because of it.

Business-to-Business Issues

Business-to-business e-commerce is very different than business-to-consumer. Business-to-business e-commerce is normally established between two organizations that have some type of relationship. One organization is normally purchasing products or services from the other. Since the two organizations have a relationship, security issues can be handled out of band (meaning that the two organizations do not have to negotiate the security issues while performing the transaction).

Availability issues may be more stringent, on the other hand. Organizations set up this type of e-commerce to speed up the ordering process and to reduce overall costs in processing paper purchase orders and invoices. Therefore, when one organization needs to make an order, the other organization must be able to receive and process it. Some business-to-business relationships will set particular times of day when transactions will take place. Others may have transactions that occur at any time.

As an example of this type of e-commerce, take an equipment manufacturer. This manufacturer uses large amounts of steel in its products and has decided to create a relationship with a local steel provider. In order to reduce inventory costs, the manufacturer wants to order steel twice a day and

have the steel delivered 24 hours after ordering for immediate use in its products. The relationship between the manufacturer and the steel mill is established so that the manufacturer will order each morning and each afternoon. That means that the steel mill's e-commerce site must be up and working properly at these times. If it is not, the manufacturer will not be able to order steel and may run out before the steel it needs is delivered. The supplier may not be able to dictate when the system must be available.

Obviously, there is an alternative if the site is down. The manufacturer could order the steel by making a phone call. Or the steel mill might see the site is down and call the manufacturer to get the order. In any case, other systems have to be employed to determine that something is not working and to use an alternative approach.

Global Time

E-commerce availability is governed by the concept of global time. This concept identifies the global nature of the Internet and of e-commerce. Traditional commerce depends upon people. People must open a store and wait for customers. The store is open during the hours that the customers are likely to be awake and shopping.

When mail order shopping was created, we began to see the concept of global time appear. Customers may choose to order products over the phone at times when they will not go out to a store. This caused mail order organizations to have employees manning the phones over a greater time period. Some mail order organizations can accept orders 24 hours a day.

The Internet is the same way. It exists all over the world. Therefore, no matter what time it is, it is daylight somewhere. Some organizations may target their products to a local audience. But just because the product is targeted at a local audience does not mean that only a local audience will be interested. Orders may come from places that were not anticipated. In order to expand the market for the organization's products, the e-commerce site must be able to handle orders from unexpected locations.

Client Comfort

In the end, availability addresses client comfort. How comfortable is the client with the ability of the organization to process the order and deliver the goods? If the site is unavailable when the customer wants to order goods, the customer is unlikely to feel comfortable with the organization.

The same is true if the customer wants to check the status of an order or to track a purchase. If the capability is advertised and is not available or does not work as advertised, the customer will lose confidence and comfort. I had this happen to me a few years ago. I ordered a software package from an online retailer. The retailer had the best price and was a well-known name. When the package did not arrive as expected, I tried to track the package via the e-commerce site.

The site advertised a way to track orders, but the function did not work. In the end, the retailer lost future business because they could not provide a simple service like accurately tracking my order.

Customer comfort or discomfort can also multiply quickly. Information is shared over the Internet in many ways, including sites that review companies and products, electronic mail lists where people discuss any number of topics, chat rooms that do the same, and news that provides a bulletin board type of discussion. Organizations that provide good service are identified on these sites and lists. People recommend these organizations to their friends and acquaintances. Organizations that do not provide good service are just as quickly identified so that the cost of failing with one customer can be multiplied hundreds if not thousands of times in minutes.

Cost of Downtime

After all this talk of the issues surrounding availability, it becomes clear that the cost of downtime is high. This cost is incurred regardless of why the e-commerce site is down. It could be hardware or software failure, a hacker causing a denial-of-service attack, or simple equipment maintenance.

The cost of downtime can be measured by taking the average number of transactions over a period of time and the revenue of the average transaction. However, this may not identify the total cost as there may be potential customers who do not even visit the site due to a report from a friend or online acquaintance. For this reason, each e-commerce site should be architected to remove single points of failure. Each e-commerce site should also have procedures for updating hardware and software that allow the site to continue operation while the systems are updated.

Solving the Availability Problem

We have discussed a lot of availability issues, but how can they be solved? The short answer is that they can't. There is no way to completely guarantee the availability of the e-commerce site. That said, there are things that can be done to manage the risk of the site being unavailable.

Before any of these management solutions can be implemented, you must decide how much the availability of the site is worth. Fail-over and recovery solutions can get expensive very quickly and the organization needs to understand the cost of the site being unavailable before an appropriate solution can be designed and implemented.

The way to reduce downtime is redundancy. We start with the communications system. If you look back at Module 16, we talked about several Internet architectures. At the very least, the Internet architecture for an e-commerce site should have two connections to an ISP. For large sites, multiple ISPs and even multiple facilities may be required.

Computer systems will house the e-commerce Web server, the application software, and the database server. Each of these systems is a single point of failure. If the availability

of the site is important, each of these systems should be redundant. For sites that expect large amounts of traffic, load-balancing application layer switches can be used in front of the Web servers to hide single failures from the customers.

When fail-over systems are considered, don't forget network infrastructure components such as firewalls, routers, and switches. Each of these may provide single points of failure in the network that can easily bring down a site. These components must also be configured to fail-over if high availability is required.

CRITICAL SKILL 17.3

Implement Client-Side Security

Client-side security deals with the security from the customer's desktop system to the e-commerce server. This part of the system includes the customer's computer and browser software and the communications link to the server (see Figure 17-1).

Within this part of the system, we have several issues:

- The protection of information in transit between the customer's system and the server
- The protection of information that is saved to the customer's system
- The protection of the fact that a particular customer made a particular order

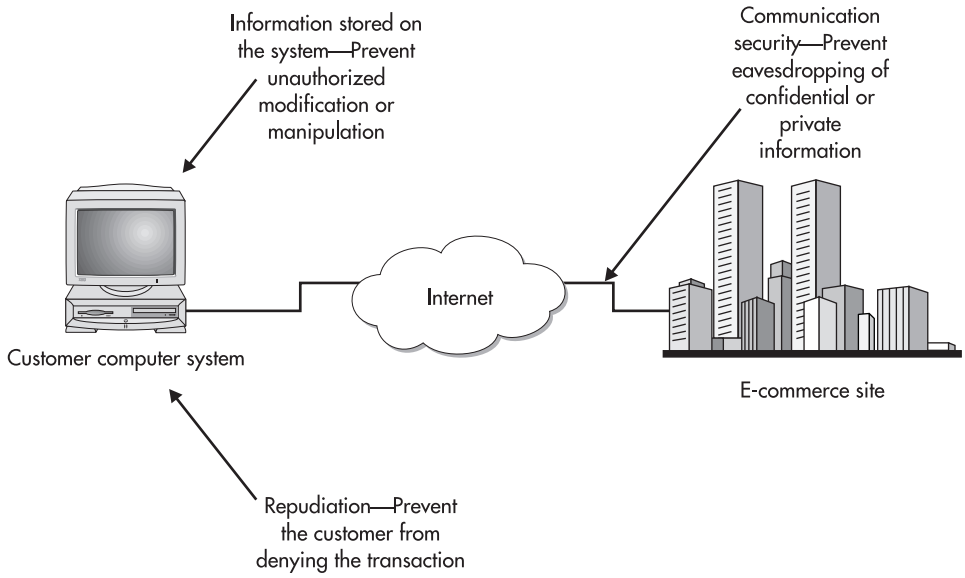

Figure 17-1 Client-side security components

Communications Security

Communications security for e-commerce applications covers the security of information that is sent between the customer's system and the e-commerce server. This may include sensitive information such as credit card numbers or site passwords. It may also include confidential information that is sent from the server to the customer's system such as customer files.

There is one realistic solution to this: encryption. Most standard Web browsers include the ability to encrypt traffic. This is the default solution if HTTPS is used rather than HTTP. When HTTPS is used, a Secure Socket Layer (SSL) connection is made between the client and the server. All traffic over this connection is encrypted.

The encryption of HTTPS will protect the information from the time it leaves the customer's computer until the time it reaches the Web server. The use of HTTPS has become required as the public has learned of the dangers of someone gaining access to a credit card number on the Internet. The reality of the situation is that consumers have a liability of at most $50 if their card number is stolen.

Saving Information on the Client System

HTTP and HTTPS are protocols that do not keep state. This means that after a Web page is loaded to the browser, the server does not remember that it just loaded that page to that browser. In order to conduct commerce across the Internet using Web browsers and Web servers, the servers must remember what the consumer is doing (this includes information

Ask the Expert

Q: **Is there any difference between 40-bit and 128-bit encryption when it comes to use in e-commerce?**

A: Module 12 has a more detailed discussion on encryption algorithms and key length. The SSL key can be 40 or 128 bits in length. The length of the key directly affects the time and effort required to perform a brute-force attack against the encrypted traffic and thus gain access to the information. Given the risks associated with sending sensitive information over the Internet, it is certainly a good idea to use encryption. However, unless the information is extremely important, there is little difference in risk between using the 40-bit or 128-bit version. For an attacker to gain access to the information, she would have to capture all of the traffic in the connection, and use sufficient computing power to attempt all possible encryption keys in a relatively short period of time (to be useful, this process cannot take years!). An attacker with the resources to do this will likely attack a weaker point such as the target's trash or perhaps the target's wallet if the credit card number is the information that is sought.

about the consumer, what they are ordering, and any passwords the consumer may have used to access secured pages). One way (and the most common way) that a Web server can do this is to use cookies.

A *cookie* is a small amount of information that is stored on the client system by the Web server. Only the Web server that placed the cookie is supposed to retrieve it, and the cookie should expire after some period of time (usually less than a year). Cookies can be in cleartext or they can be encrypted. They can also be persistent (meaning they remain after the client closes the browser) or they can be non-persistent (meaning they are not written to disk but remain in memory while the browser is open).

Cookies can be used to track anything for the Web server. One site may use cookies to track a customer's order as the customer chooses different items. Another site may use cookies to track a customer's authentication information so that the customer does not have to log in to every page.

The risk of using cookies comes from the ability of the customer (or someone else with access to the customer's computer) to see what is in the cookie. If the cookie includes passwords or other authentication information, this may allow an unauthorized individual to gain access to a site. Alternatively, if the cookie includes information about a customer's order (such as quantities and prices), the customer may be able to change the prices on the items.

When an order is placed, the prices should be checked if stored in a cookie.

The risk here can be managed through the use of encrypted and non-persistent cookies. If the customer order or authentication information is kept in a non-persistent cookie, it is not written to the client system disk. An attacker could still gain access to this information by placing a proxy system between the client and the server and thus capture the cookie information (and modify it). If the cookies are also encrypted, this type of capture is not possible.

Repudiation

One other risk associated with the client side of e-commerce is the potential for a client or customer to repudiate a transaction. Obviously, if the customer truly did not initiate the transaction, the organization should not allow it. However, how does the organization decide whether a customer is really who he says he is? The answer is through authentication.

The type of authentication that is used to verify the identity of the customer depends on the risk to the organization of making a mistake. In the case of a credit card purchase, there are established procedures for performing a credit card transaction when the card is not present. These include having the customer provide a proper mailing address for the purchase.

If the e-commerce site is providing a service that requires verification of identity to access certain information, a credit card may not be appropriate. It may be better for the organization to use user IDs and passwords or even two-factor authentication. In any of these cases, the terms of service that are sent to the customer should detail the requirements for protecting the ID and password. If the correct ID and password are used to access customer information, it will be assumed by the organization that a legitimate customer is accessing the information. If the password is lost, forgotten, or compromised, the organization should be contacted immediately.

Progress Check

1. The two primary differences between e-commerce services and traditional Internet services are the need for _____ and _____.
2. Availability is very important because it directly leads to the issue of _____, which will help a customer determine if they will purchase from you or a competitor.

CRITICAL SKILL 17.4

Implement Server-Side Security

When we talk about server-side security, we are only talking about the physical e-commerce server and the Web server software running on it. We will examine the security of the application and the database in the next sections of this module. The e-commerce server itself must be available from the Internet. Access to the system may be limited (if the e-commerce server only handles a small audience) or it may be open to the public.

There are two issues related to server security:

- The security of information stored on the server
- The protection of the server itself from compromise

Information Stored on the Server

The e-commerce server is open to access from the Internet in some way. Therefore, the server is at most semi-trusted. A semi-trusted or untrusted system should not store sensitive information. If the server is used to accept credit card transactions, the card numbers should be immediately removed to the system that actually processes the transactions (and that is located in a more secure part of the network). No card numbers should be kept on the server.

1. Authentication and confidentiality
2. Client comfort

If information must be kept on the e-commerce server, it should be protected from unauthorized access. The way to do this on the server is through the use of file access controls. In addition, if the sensitive files are not stored within the Web server or FTP server directory structure, they are much harder to access via a browser or FTP client.

Protecting the Server from Attack

The e-commerce server will likely be a Web server. As mentioned before, this server must be accessible from the Internet and therefore is open to attack. There are things that can be done to protect the server itself from successful penetration. These things fall into three categories:

- Server location
- Operating system configuration
- Web server configuration

Let's take a closer look at each of these.

Server Location

When we talk about the location of the server, we must talk about its physical location and its network location. Physically, this server is important to your organization. Therefore, it should be located within a protected area such as a data center. If your organization chooses to place the server at a co-location facility, the physical access to the server should be protected by a locked cage and separated from the other clients of the co-location facility.

When choosing a co-location facility, it is good practice to review their security procedures. In performing this task for clients, my team and I have found that many sites do have good procedures but poor practice. While performing inspections at co-location facilities, we have been able to gain access to cages we were not authorized to enter. At times this access has been facilitated by the guard who was escorting us.

The network location of the server is also important. Figure 17-2 shows the proper location of the server within the DMZ. The firewall should be configured to only allow access to the e-commerce server on ports 80 (for HTTP) and 443 (for HTTPS). No other services are necessary for the public to access the e-commerce server and therefore should be blocked at the firewall.

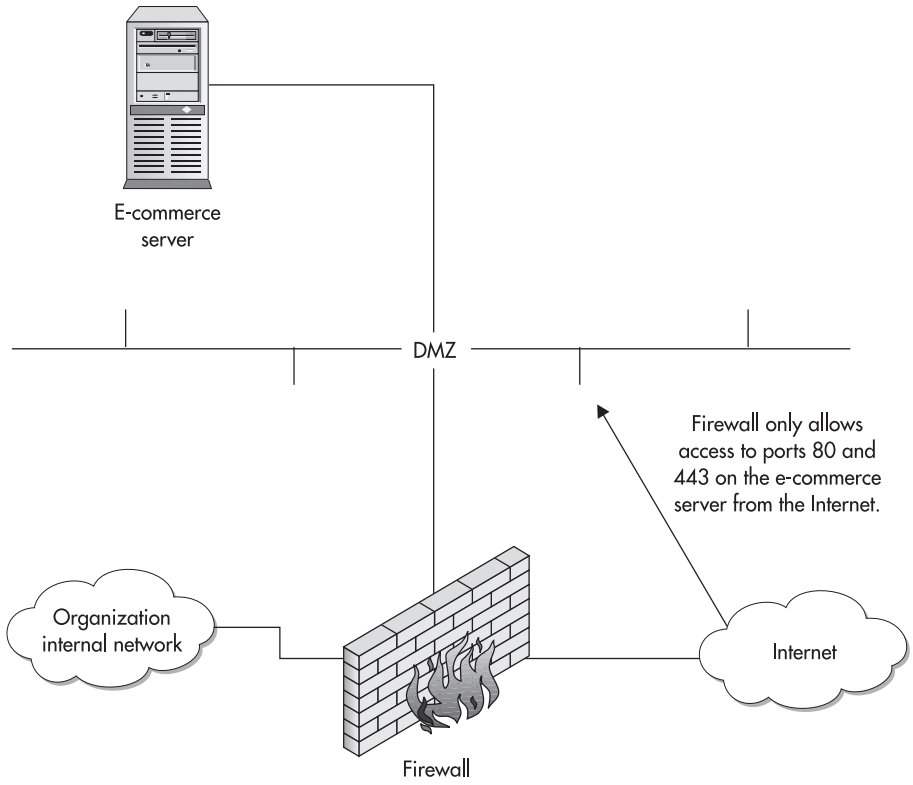

Figure 17-2 Proper network location for the e-commerce server

If performance of the e-commerce server is extremely important and traffic to the server is expected to be very high, it may be appropriate to dual-home the server (see Figure 17-3). In this case one network interface handles the incoming Web traffic and sends responses to the customer. This interface resides on the DMZ. The second network interface handles application queries either to an application server (the preferred architecture) or directly to the back-end database. This second interface resides on a second DMZ or application server network. This network is also separated from the organization's internal network by a firewall. It is never a good idea to have one interface on the Internet and one interface on the internal network.

Operating System Configuration

The e-commerce server operating system should be configured with security in mind. The choice of operating system depends on a number of factors, including the expertise of the organization's administration staff. In today's world, the primary operating system choices are Unix or Windows 2000. Both operating systems can be configured in a secure manner and

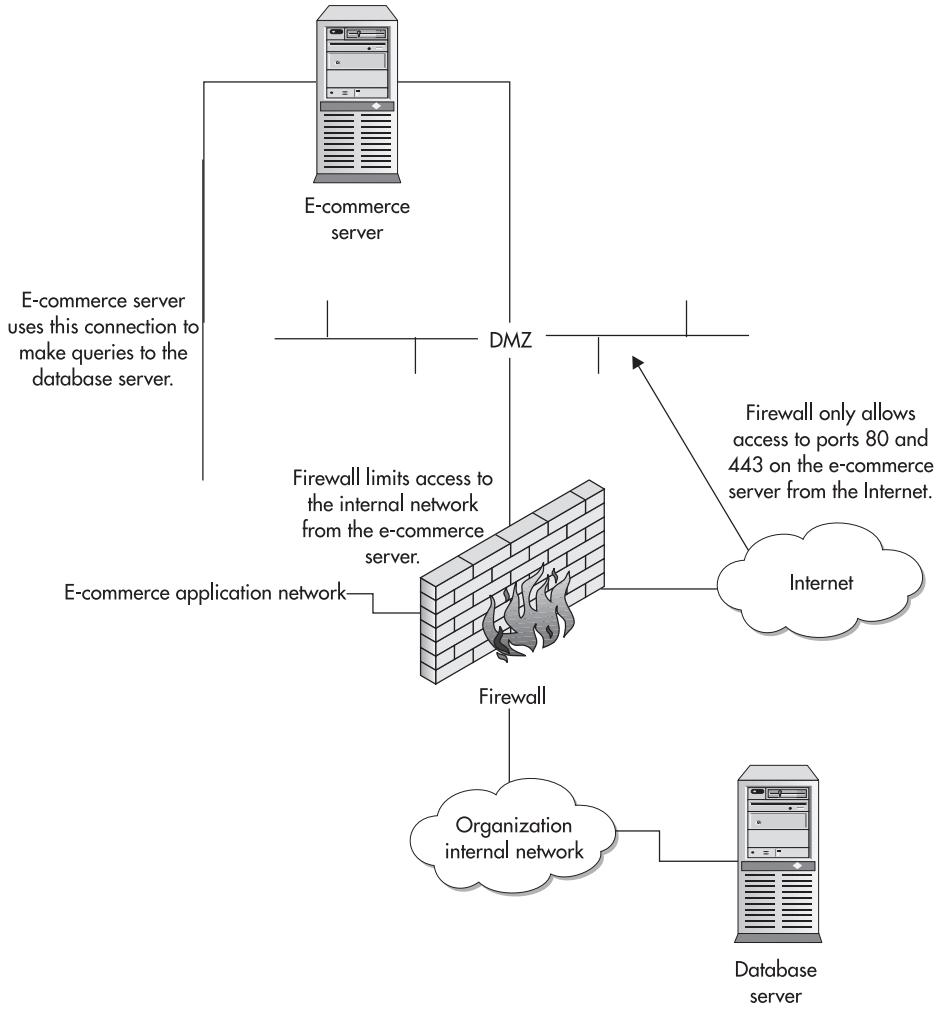

Figure 17-3 E-commerce server location when two network interfaces are required

both can also be configured in an insecure manner. When choosing the operating system, other factors such as performance requirements and fail-over capabilities must be considered. Also, it is better to choose an operating system that the administration staff is familiar with rather than one that is unfamiliar.

The first step in configuring the server securely is to remove or turn off any unnecessary services. The system is primarily a Web server and, therefore, it must run a Web server. Does the system really need to run DNS? Probably not, so turn it off. Go through the services that are running on the system and identify those that are necessary for the operation of the system. Turn off any that are not required.

The next step is to patch the system. Check for the latest patches for the chosen operating system and load them. Once the patches are loaded, configure the system to conform to organization policy with regard to password length and change frequency, audit, and other requirements.

When downloading patches for the chosen operating system, don't just download the current patch cluster. Some manufacturers separate security patches from the main patch cluster. If the security patches are not specifically downloaded, the system will not be patched properly.

Before the system is declared ready for production, you should scan it for vulnerabilities. Vulnerability scanners can be commercial or freely available, but they must be current. Check the system to confirm that you have turned off all unnecessary services and loaded all necessary patches. This scan will confirm that the system is currently free from vulnerabilities. Scans should be performed on a monthly basis with the latest updates to the scanners to make sure the system is still free from vulnerabilities. New vulnerabilities that are found should be fixed immediately.

Web Server Configuration

The Web server itself is the last component of the server security. Many Web servers are available on the market and the choice of which server to use will depend on the platform chosen and the preferences of the administration and development staffs. As with operating systems, Web servers can be configured in a secure manner or an insecure manner. The specific configuration requirements for each particular Web server are beyond the scope of this book, but there are some common configurations that should be made regardless of the Web server. First, the server software should be upgraded and patched according to the manufacturer's recommendations.

Never run the Web server as root or administrator. If the Web server is successfully penetrated, the attacker will have privileges on the system the same as those of the Web server. If the Web server is run as root, the attacker will have root privileges. Instead, create a separate user who owns the Web server and run the server from that account.

Each Web server requires the administrator to define a server root directory. This directory tells the Web server where to find document files and scripts and also limits the Web server in what files can be accessed via a browser. The Web server root should never be the same as the system root directory, and it should not include configuration and security files that are important to the operating system (see Figure 17-4).

Most Web servers come with CGI scripts (CGI is the Common Gateway Interface and is used for creating scripts on a Web server). Some of these default scripts have very serious vulnerabilities that allow attackers to gain access to files or the system itself. Any scripts that come with the Web server that are not being used by the Web site should be removed to prevent an attacker from using them to gain access to the system.

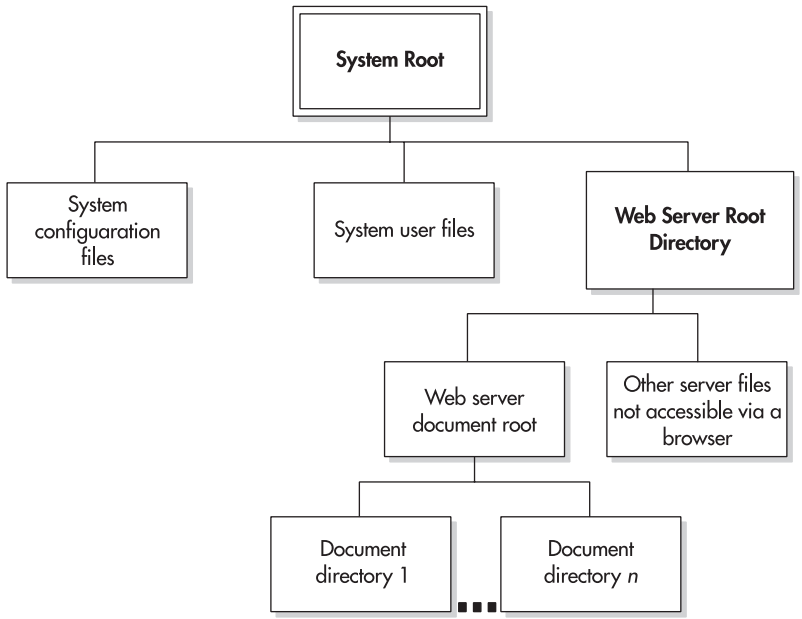

Figure 17-4 Proper Web server root directory structure

CGI scripts should not be visible to the public either. This means that the Web server should be configured not to show directory listings if the browser does not specify a file. If the browser does specify a CGI or Perl script, the server should be configured to execute the script rather than display the code. This is normally configured in the httpd.conf file with the lines:

```
AddType application/x-httpd-cgi .cgi
AddType application/x-httpd-cgi .pl
```

As with the operating system, the Web server should be scanned for known vulnerabilities before the system is placed in production. It may be possible to use the same scanner as that used for the operating system, but make sure that the scanner includes checks against the Web server. Once the system is in production, the Web scans should be conducted on the same schedule as the operating system scans.

CRITICAL SKILL 17.5

Implement Application Security

The security of the e-commerce application as a whole is perhaps the most important part of e-commerce security. The application also includes the procedures for handling operations such as page changes and software upgrades.

Proper Application Design

Let's start the discussion of application security with the design of the application itself. When an e-commerce application is being designed, an organization should perform the same project steps as the design and development of any large, complex system, namely:

- Requirements definition
- System design
- Development
- Testing
- Deployment

All of these steps should be laid out in the organization's development manual.

Security requirements should be included in the requirements definition phase of the project. Security requirements that should be specified include

- Identification of sensitive information
- Protection requirements for sensitive information
- Authentication requirements for access or operations
- Audit requirements
- Availability requirements

If these requirements have been defined, then when the system design phase begins, we can identify potential design issues. All sensitive information should be protected in some manner. This will govern what parts of the application require HTTPS vs. HTTP. Sensitive information may not require only encryption in transit. Some information, such as private information about the customer, may require protection when written to the customer's computer system in cookies. The design should take this into account and in this case use encrypted cookies.

One other issue about sensitive information should be mentioned here. Information may be sensitive because of the way the application will use the information. For example, some applications pass information between programs using the URL (universal resource locator or the Web site address in the browser). If you see a long URL with "?" separating various values, the application is passing parameters to other scripts or programs. The customer can change these parameters and thus adjust the way the programs behave. Some e-commerce sites record customers' purchasing choices in the URLs. The information that is recorded in these URLs includes the item number, quantity, and price. If the price is not checked on the back end of the process, customers could change the prices of items. In one case, a customer changed

the price to a negative number and the organization provided a credit to the customer for each item purchased. Given this example, it becomes clear that the prices of items may be sensitive to the organization. If the URL is used to pass this information between scripts or programs, the prices (at least) should be checked at the back end before the order is processed.

Sensitive information such as credit card numbers may also be stored by the organization. As mentioned before, it is never a good idea to store such valuable information on the Web server itself. The system design should provide a mechanism for getting this information off the Web server and either store it in the database server or delete it after it has been used. When deciding whether to keep credit card information or not, one consideration is how the customer feels. Some marketing groups say that a customer wants the e-commerce process to be as easy and painless as possible and that retyping credit card numbers may cause customers to go to a different site, so this may be a requirement. If it is, the card numbers must be kept someplace where the risk of a successful attack is small.

Along these same lines, the organization may choose to avoid this issue entirely by using an outside partner to process the credit card transactions. If this option is chosen, the information on the purchase must be handed off to the partner. Care must be taken here to pass the information correctly.

Proper Programming Techniques

Any e-commerce application will require some coding either of scripts or programs. These are likely to be custom programs designed specifically for your particular environment and situation. The programs are a major source of system vulnerabilities primarily due to programming errors. The biggest of these errors is the potential for buffer overflows. Buffer overflow problems can be reduced by correcting two errors:

- Do not make assumptions about the size of user input.
- Do not pass unchecked user input to shell commands.

If the programmer makes assumptions about the size of expected user input, he is likely to define particular variable sizes. If an attacker knows this, she might be able to send input that will cause the input buffer to overflow and potentially gain access to files or the operating system (see Module 3 for a more detailed discussion of buffer overflows).

The second issue is a more specific subset of the first issue. If your programs make calls to shell commands, user input should not be blindly passed to the shell command. The user input should be verified to make sure that it is appropriate for the command.

Many of these errors can be caught before the site goes into production if the code is subjected to a peer review or a code review. Unfortunately, few development projects seem to budget enough time for this type of activity. At the very least, the development staff should be given a security briefing about these types of errors prior to the start of the coding effort.

To more completely evaluate the vulnerabilities on the site, instead of using only a system vulnerability scanner, use an application scanner as well to look for vulnerabilities. One such commercial tool is WebInspect from SPI Dynamics (**http://www.spidynamics.com/**).

Showing Code to the World

Vulnerability scanners should detect buffer overflow problems in well-known programs and scripts before the site goes into the production. This step is critical since these vulnerabilities are known to the hacker community and thus may be used to attack your site. Overflow problems in custom code will not be known to the hacker community and thus may not be easily found by an attacker. However, if an attacker is very interested in penetrating your e-commerce site, he will examine all of the information he can in order to find a vulnerability.

One step that he may take to do this is to examine your scripts via your Web site. Proper Web server configuration should limit his ability to do this, but if the scripts exist on the site, there may be a configuration mistake that allows him to see the scripts. Another option to prevent this type of examination is to write the entire application in a compiled language such as C or C++ rather than in an interpreted language such as CGI or Perl.

Configuration Management

Once the application has been written and tested, it will be moved into production and opened up to the world. If you have followed good security practice to this point, you have taken significant numbers of precautions with your site. Now is not the time to stop working on security. One last item must be attended to and that is configuration management. There are two parts of configuration management:

- The control of authorized changes
- The identification of unauthorized changes

The control of authorized changes is done with procedures and policy. Only certain employees will be authorized to make changes to programs or Web pages. Before updates to programs should be moved into the production, they should be tested on a development or quality control system. Changes to Web pages should also go through a quality control process to detect spelling and grammar errors.

Development and testing should take place on a separate system that mimics the production system. No development or "fixes" should take place on the production system.

The identification of unauthorized changes should be a part of any system that displays your organization to the world. The e-commerce site is a prime example of this. Each program component (script or compiled program) and each static Web page should be constantly checked for an unauthorized change. The most common way to do this is via a cryptographic checksum (more detail on this can be found in Module 12). When a file is placed on the production system, a checksum should be run on it. Periodically after that a checksum should be run and compared with the original. If they differ, an alert should be created so the system can be examined for a successful penetration. In extreme cases, the program that performs the check could reload a copy of the original file. To prevent false alarms, an update of the checksum should be part of the configuration management procedure.

CRITICAL SKILL 17.6

Implement Database Server Security

To complete the design of security for electronic commerce, we must also address the database server that holds all of the e-commerce transactions. Somewhere in the depths of the organization's network there will have to exist a database into which all of the customer information, order information, shipping information, and transaction information will eventually find its way. This database contains a lot of sensitive information. The information in the database may be confidential in nature, thus requiring some confidentiality protection, or it may be sensitive because it must be accurate, thus requiring integrity protection. The server may also form a key component in the e-commerce system and may require availability protection as well.

Given the sensitivity of the information in the database, the following issues must be examined:

- The location of the database server
- How the database server communicates with the Web server or application server
- How the database server is protected from internal users

Database Location

As with the Web server, the physical location of the system should be someplace where access can be controlled. The data center is a good location. While the database server could be located at a co-location facility, the sensitive nature of the information contained in the database means that it should be located in a facility completely under the control of the organization.

The best network location for the database server is in the organization's internal network. Since there is no reason for the database server to be accessed by anyone external to the organization, it does not need to be connected to the Internet. It is a completely trusted system as well so it does not introduce additional risk to the internal network by residing there.

In some cases, the database server is so sensitive that it is placed in a separate part of the network. This part of the network is protected by an internal firewall, and traffic through the firewall is severely limited.

Communication with the E-Commerce Server

The database server must communicate with the e-commerce server so that transactions may be processed. Normally, this communication is via a SQL connection (see Figure 17-3). In the best of all possible worlds, the database server will initiate the connection to the system in the DMZ. This is ideal because the DMZ system is in an untrusted part of the network and should not be making connections to the internal or trusted part of the network. However, this requires the e-commerce server to store transaction information (and possibly queries as well) until the database server initiates the connection. This may delay transactions or the providing of information to the customer. In most cases, this is unacceptable to the organization.

The only alternative is for the e-commerce server to initiate the SQL connection to the database server. This brings up a number of security issues. First, the e-commerce server must have an ID and password to the database server in order to do this. This ID and password must be embedded in a program or written to a file on the system. If the ID and password exist on the e-commerce system, an intruder could learn the ID and password and potentially gain access to the database server. Since the database server contains sensitive information, this is not a good thing to have happen.

One way around this issue is to make the ID and password used by the e-commerce server a very restricted ID. The ID would have access to send transaction information to a single table (write access), but it would not have read access to any tables in the database. This configuration works fine for some applications, but it does not allow the e-commerce server to get information to present to a customer. If this is necessary, the ID could be granted read access to non-sensitive information in the database, such as catalog information, so it can be queried and presented to the customer.

What if the information that needs to be presented is sensitive? This presents a big problem. For example, what if a bank customer wants to query an account balance? How can this be handled? In the best case, the ID and password that exist on the e-commerce server would be coupled with some form of authentication provided by the customer in order to release the information. That way, if an attacker did penetrate the e-commerce server, he would not be able to gain access to sensitive customer information.

The risk can be further reduced in this case by dividing the functionality of the e-commerce server between a Web server and an application server. The Web server presents the information to the customer and accepts information from the customer. The application server processes the information from the customer, makes queries to the database server, and provides information to the Web server for presentation to the customer (see Figure 17-5).

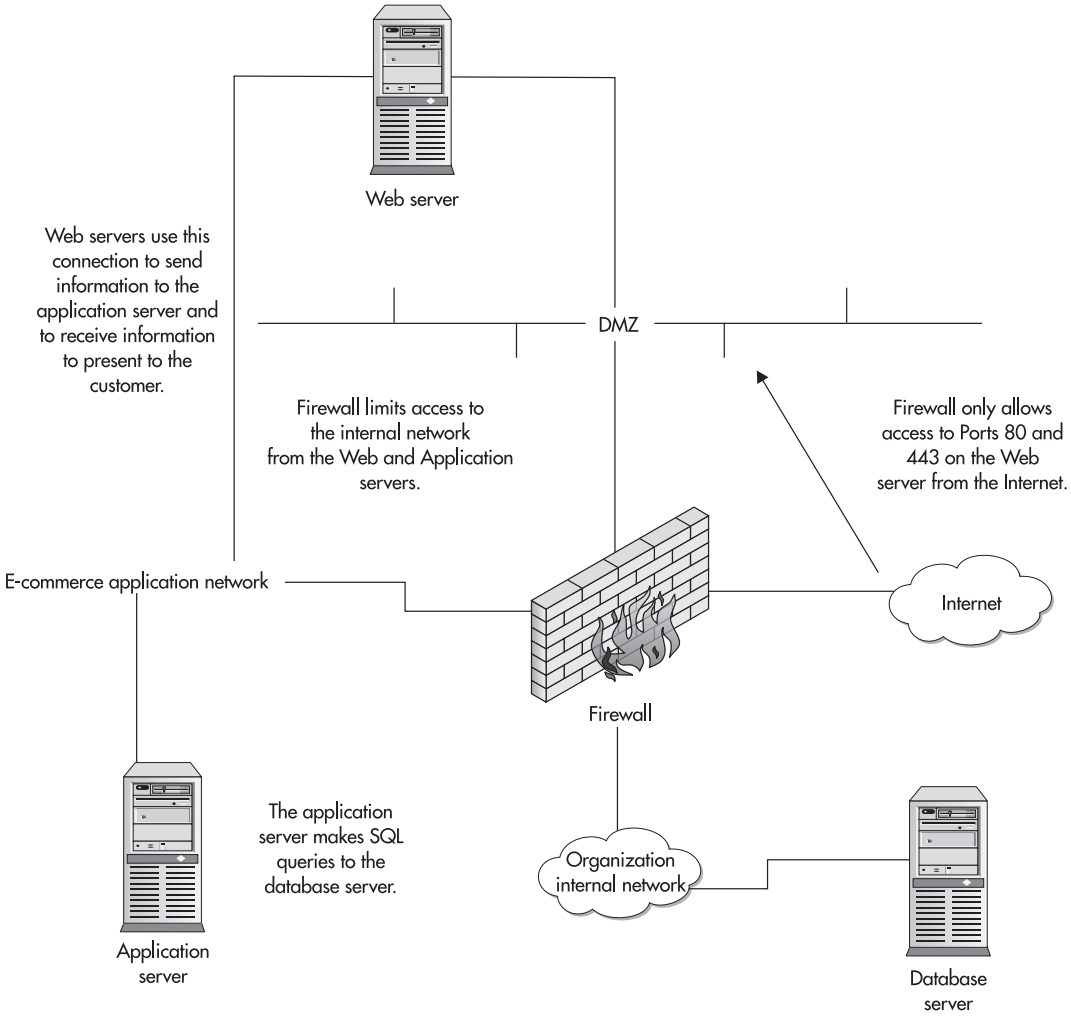

Figure 17-5 Revised e-commerce architecture using an application server

Internal Access Protection

All of the security issues that we have discussed so far have been related to external threats. Unfortunately, they are not the only threats that must be examined. The database server contains sensitive information. Employees of the organization have access to the internal

network where the database server resides and therefore have the ability to directly attack it without having to work through a firewall and Web server first.

One solution to this problem was mentioned above. The database server could be moved to a separate network and protected by an internal firewall. This is not the only solution. The server itself should be scanned for vulnerabilities on the same schedule as the Web server. It should be patched before going into production, and IDs and passwords should be controlled as defined in organization policy.

In addition, the database should be configured to audit access attempts to it.

Databases offer an attacker the ability to gain access to information without accessing the underlying operating system. In order to properly watch the system for access attempts and attempted vulnerability exploits, the operating system logs and the database logs must both be watched.

Given the sensitivity of the information in the database, authorized access to the system should be controlled. The system should not be a general use system, and development should not be allowed on the system.

CRITICAL SKILL 17.7

Develop an E-Commerce Architecture

Let's put everything together. Figure 17-6 provides a diagram of a total e-commerce site. The figure includes architectural components for a full-up, high-traffic, high-availability site. Depending on the amount of traffic and your security requirements, some of these components may not be necessary.

Server Location and Connectivity

This is a high-traffic, high-availability e-commerce site. Therefore, the organization has links with two different ISPs, and the ISPs have agreed to run BGP between them so that fail-over routing is established. In this case, we are assuming that the organization has chosen to place all of its e-commerce servers at a single facility. This architecture could be expanded to include other facilities.

The routers, switches, and firewalls connected to the Internet are cross-connected so that the failure of any one component will not affect the traffic to the site. Behind the firewalls, two application layer switches handle load balancing across the Web servers. The Web servers are protected from attack on all ports other than 80 and 443 by the firewalls.

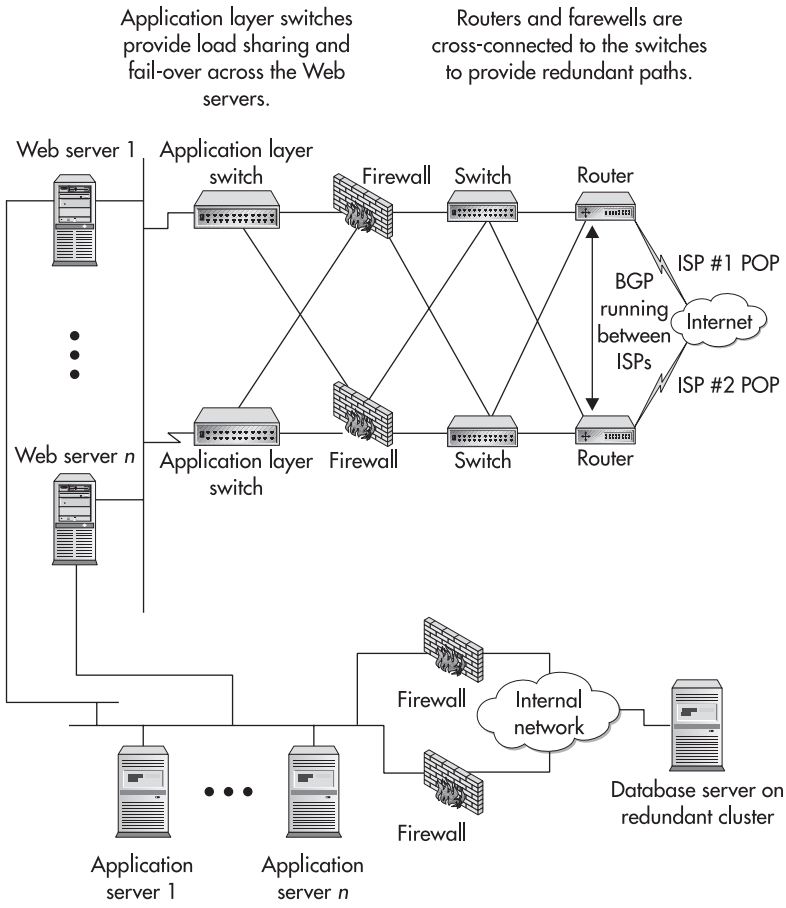

Figure 17-6 E-commerce architecture for a high-availability site

The Web servers have a second network interface that connects to a network where the application servers reside. The Web servers pass information to the application servers that query the database and pass information for the customer back to the Web servers. Dual firewalls connect the application server network to the organization's internal network where the database server resides.

Availability

As you can see from Figure 17-6, there is no single point of failure in this design. The application server network may also consist of redundant switches so that there is always an available path from the customer to a Web server to an application server to the database server. The cost of this availability is more than double the cost of a basic Internet site. Not only does this design require at least two of all network components and servers, but it also adds the application layer switches to the design. Depending on the traffic load, the number of Web servers and application servers may be large (greater than 20 of each, for example). This will also necessitate that the database server be able to handle a large number of transactions per second.

For sites where latency is a key factor, the front-end firewalls may be removed. While this is not a wise security decision, it may be necessary to meet the latency requirements for the site. In this case, the routers should be configured to filter all traffic other than ports 80 and 443.

Vulnerability Scanning

A regular program exists to scan all of the systems from time to time. Scans are performed from four locations:

- Outside the firewall to see what ports are allowed through the firewall and what vulnerabilities can be seen from the Internet
- On the Web server network to detect the services and vulnerabilities on the Web servers
- On the application server network to detect the services and vulnerabilities on the Web server's second interface and on the application servers
- On the organization's internal network to detect services and vulnerabilities on the database server

These scans are conducted on a monthly basis and the correction of vulnerabilities is tracked. New systems are scanned before being brought into production.

Audit Information and Problem Detection

Audit trails are captured on the database server and examined to detect internal employees who might be attempting to make changes to the database. Key files on the Web servers and application servers are checked for changes every ten minutes to quickly detect systems that may have been compromised.

Project 17 Design an E-Commerce Architecture

This project is intended to take you through the steps of designing a site for e-commerce. For this project, we will assume that a bank wants to establish a home banking system for its customers. The bank already has a data center with the appropriate physical security. All customer account information is stored on a mainframe computer. Each customer already has a PIN that is used at automated cash machines.

The bank wants to offer customers access to their accounts for the following activities:

- Transfers of funds between accounts at the bank
- Ordering checks
- Checking account balances and examining recent transactions
- Bill payment via a partner (the customer will be redirected to the partner Web site for this with no additional login)

Step by Step

1. Begin by defining the security requirements for the system in each of the four security services: confidentiality, integrity, availability, and authentication.
2. Determine a high-level system design that meets the security requirements. For this part of the system, assume that the system will interact with the mainframe to get customer account information and to perform transfers and check orders.
3. Define specific security requirements on each system component: client system, Web server, application, and database.
4. Define the overall architecture of the system, including what components are needed to protect each system. Do not assume that network security components exist in the network. Instead, identify all of the required components.
5. Add to this design the necessary additional systems to meet the availability requirements.

Project Summary

This project is a large design project that usually includes the efforts of a number of people. Remember to focus primarily on the security aspects of the design. This will give you a better idea of what the design process is all about. To do this design work properly, you must assess the risk to the bank and identify proper security countermeasures to manage the risk.

Module 17 Mastery Check

1. What is the most critical security service for e-commerce?
2. Generally speaking, which type of e-commerce has greater issues with uptime?
3. What is meant by "global time"?
4. Can the cost of downtime be measured directly?
5. If information must be stored on the client system, what should be used to protect the confidentiality of the information?
6. In an e-commerce site, where should customer information be kept?
7. Where should e-commerce servers that interact with the customer be located?
8. When configuring a Web server, where should the Web pages reside?
9. In what file should .cgi and .pl files be defined so that the programs are run and the source code not shown as a Web page?
10. If sensitive information is involved in a transaction, what is the best location to use to store session information?
11. During the development phase of the project, developers should prevent buffer overflows by not passing user input directly to shell commands and _____.
12. In a three-tiered e-commerce architecture, does the database server have any contact with the front end Web servers?
13. What types of vulnerability scans should be made on e-commerce sites?
14. What type of system is best to identify configuration control problems?
15. Can availability be completely assured by redundant equipment?