CHAPTER

15 Firewalls

Firewalls have been one of the most popular and important tools used to secure networks since the early days of interconnected computers. The basic function of a firewall is to screen network traffic for the purposes of preventing unauthorized access between computer networks.

What generates that traffic? Applications. Running internally and externally on servers and workstations (and sometimes on other network devices or appliances), applications are the purpose of all network traffic. Thus, what we're really talking about when we're discussing firewalls is application communication management on layers one through seven of the OSI stack (which is described in Chapter 14). Applications are what firewalls are really all about. Don't think of the firewall as just a network appliance—think of it as one of the tools used for managing the behavior of applications.

Overview

Firewalls are the first line of defense between the internal network and untrusted networks like the Internet. You should think about firewalls in terms of what you really need to protect, so you will achieve the right level of protection for your environment.

First introduced conceptually in the late 1980s in a whitepaper from Digital Equipment Corporation, "firewalls" provided a then new and important function to the rapidly growing networks of the day. Before dedicated hardware was commercially available, router-based access control lists were used to provide basic protection and segregation for networks. However, they proved to be inadequate as emerging malware and hacking techniques rapidly developed. Consequently, firewalls evolved over time so their functionality moved up the OSI stack from layer three to layer seven.

The Evolution of Firewalls

First-generation firewalls were simply permit/deny engines for layer three traffic, working much like a purposed access control list appliance. Originally, first-generation firewalls were primarily used as header-based packet filters, capable of understanding source and destination information up to OSI layer four (ports). However, they could not perform any "intelligent" operations on the traffic other than "allow or deny it from this predefined source IP address to this predefined destination IP address on these predefined TCP and UDP ports."

Second-generation firewalls were able to keep track of active network sessions, putting their functionality effectively at layer four. These were referred to as *stateful firewalls* or, less commonly, *circuit gateways*. When an IP address (for example, a desktop computer) connected to another IP address (say, a web server) on a specific TCP or UDP port, the firewall would enter these identifying characteristics into a table in its memory. This allowed the firewall to keep track of network sessions, which could give it the capability to block *man-in-the-middle (MITM)* attacks from other IP addresses. In some sophisticated firewalls, a high-availability (HA) pair could swap session tables so that if one firewall failed, a network session could resume through the other firewall.

The third generation of firewalls ventured into the application layer—layer seven. These "application firewalls" were able to decode data inside network traffic streams for certain well-defined, preconfigured applications such as HTTP (the language of the web), DNS (the protocol for IP address lookups), and older, person-to-computer protocols such as FTP and Telnet. Generally, they were unable to decrypt traffic, so they were unable to check protocols like HTTPS and SSH. They were designed with the World Wide Web in mind, which made them well suited to detecting and blocking web site attacks that were generating a great deal of concern at the time, like cross-site scripting and SQL injection.

Consider these in comparison to today's current generation of firewalls (commonly termed the fourth generation), which have the intelligence and capability to look inside packet payloads and understand how applications function. As silicon has increased in speed, advanced router-based firewalls exist today that can provide IP inspection as a software component of a multipurpose router, although they do not provide the speed or sophistication of today's industrial-strength firewalling solutions. In addition, unified threat management (UTM) devices have combined sophisticated, application-layer firewalling capability with antivirus, intrusion detection and prevention, network content filtering, and other security functions. These are true layer seven devices.

Fourth-generation firewalls can run application-layer gateways, which are specifically designed to understand how a particular application should function and how its traffic should be constructed and patterned (traffic that conforms predictably to an application's well-defined communication protocol is referred to as "well formed"). There are fifth-generation firewalls, which are internal to hosts and protect the operating system kernel, and some sixth-generation firewalls have been described (meta firewalls), but most network appliances you will find today fall into the generally accepted fourth-generation firewall definition. Some manufacturers call their devices "next-generation firewalls" or "zone-based firewalls," and these essentially function under the same guiding principles of the fourth-generation designs. In this chapter, we primarily focus on fourth-generation firewalls and the key functionality that they enable.

Application Control

From the beginning, firewalls have always been intended to handle application traffic. Some applications are authorized, and some aren't. For example, web traffic outbound to Internet web sites is commonly permitted, while some types of peer-to-peer software are not. On those applications that are allowed, certain behaviors are allowed within the application and others aren't. For instance, web-based meeting and collaboration software might be approved for use on the Internet, but the file-sharing capabilities might be restricted.

First- and second-generation firewalls could restrict simple applications that functioned on well-known ports. Back then, applications were well behaved, communicating on assigned ports that were well documented, so they were easy to control. But application developers did not always want to be subject to control, so they devised a simple but effective way to get through the firewall—use port 80. This is known as "tunneling" or "circumventing." Since web traffic uses the HTTP protocol over TCP port 80, it had to be allowed to pass through the firewall unrestricted. There was no practical way to keep track of the millions of IP addresses on the Internet, so applications could freely communicate and their developers were happy.

But then application firewalls came along. These devices could observe the contents of the HTTP traffic traversing port 80, and determine whether it consisted of web site to browser requests and responses, or something else tunneling through from an application on a local workstation to a remote server. This provided a rudimentary ability to block applications that were prohibited by security policies, but it didn't usually help with controlling application behavior (such as allowing voice but not video, or transfer of document files but not photos and movies). Security administrators were concerned about different types of software that could violate security policies, such as:

- **Peer-to-peer file sharing** Direct system-to-system communication from an inside workstation to another one on the Internet that could leak confidential documents, or expose the organization to liability from music and movie copyright violations
- **Browser-based file sharing** Web sites that provide Internet file storage via a web browser, which allow trusted people inside an organization's network to copy files outside the security administrator's area of control
- **Web mail** Mail services with the capability to add file attachments to messages, providing a path to theft and leakage of confidential materials
- **Internet proxies and circumventors** Services running on the Internet or on local workstations explicitly designed to bypass security controls like web filtering
- **Remote access** Remote administration tools, usually used by system administrators to support internal systems from the Internet, which could be abused by Internet attackers

None of these were easy to control using application-aware firewalls, which could really only block broad categories of applications from functioning, or the Internet addresses they needed to connect to, but never with 100 percent effectiveness. That's where fourth-generation firewalls come in. These devices have advanced heuristic application detection and behavior management capabilities. Circumventing network security controls by using

allowed ports isn't effective any more. Until application developers come up with a new way to circumvent the firewall, the security administrator is back in control.

When Applications Encrypt

Applications that want to bypass firewalls may encrypt their traffic. This makes the firewall's job more difficult by rendering most of the communication unreadable. Blocking all encrypted traffic isn't really feasible except in highly restricted environments where security is more important than application functionality, and a "permit by exception" policy blocks all encrypted application traffic except for that on a whitelist of allowed, known applications. And broad-spectrum decryption capability isn't within the reach of most consumers and enterprises, despite Moore's law's predicted wholesale advances in computing power.

However, controlling application communications can still be done even if the traffic is encrypted, by some of the more advanced fourth-generation firewalls. Applications are easiest to identify by the unique signatures inside their data streams, but there are other identifying features as well. Most have a "handshake protocol" that governs the start of a session, and these usually have an identifiable pattern. Many also have identifiable IP addresses on the Internet they communicate with. Even traffic pattern analysis is possible with advanced heuristic capabilities. A lot of information can be gleaned just from the frequency, size, and timing of communications.

Applications that encrypt their network traffic can be controlled by fourth-generation firewalls, although it's easier to permit or deny the entire application than it is to control the specific functions within it. Today's fourth-generation firewalls have extensive lists of known applications based on extensive research and analysis ready to drag-and-drop into a policy configuration.

Must-Have Firewall Features

Today's firewalls are expected to do much more than simply block traffic based on the outward appearance of the traffic (such as the TCP or UDP port). As applications have become increasingly complex and adaptive, the firewall has become more sophisticated in an attempt to control those applications. You should expect at least the following capabilities from your firewall.

Application Awareness

The firewall must be able to process and interpret traffic at least from OSI layers three through seven. At layer three, it should be able to filter by IP address; at layer four by port; at layer five by network sessions; at layer six by data type, and, most significantly, at layer seven to properly manage the communications between applications.

Accurate Application Fingerprinting

The firewall should be able to correctly identify applications, not just based on their outward appearance, but by the internal contents of their network communications as well. Correct application identification is necessary to ensure that all applications are properly covered by the firewall policy configuration.

Granular Application Control

In addition to allowing or denying the communication among applications, the firewall also needs to be able to identify and characterize the features of applications so they can be managed appropriately. File transfer, desktop sharing, voice and video, and in-application games are examples of potentially unwanted features that the firewall should be able to control.

Bandwidth Management (QoS)

The Quality of Service (QoS) of preferred applications, which might include Voice over IP (VoIP) for example, can be managed through the firewall based on real-time network bandwidth availability. If a sporting event is broadcast live via streaming video on a popular web site, your firewall should be able to proactively limit or block access so all those people who want to watch it don't bring down your network. The firewall should integrate with other network devices to ensure the highest possible availability for the most critical services.

Core Firewall Functions

Due to their placement within the network infrastructure, firewalls are ideally situated for performing certain functions in addition to controlling application communication. These include Network Address Translation (NAT), which is the process of converting one IP address to another, and logging of traffic.

Network Address Translation (NAT)

The primary version of TCP/IP used on the Internet is version 4 (IPv4). Version 4 of TCP/IP was created with an address space of 32 bits divided into four octets, mathematically providing approximately four billion addresses. Strangely enough, this is not sufficient. A newer version of IP, called IPv6, has been developed to overcome this address-space limitation, but it is not yet in widespread deployment.

In order to conserve IPv4 addresses, RFC 1918 specifies blocks of addresses that will never be used on the Internet. These network ranges are referred to as "private" networks and are identified in Table 15-1. This allows organizations to use these blocks for their own corporate networks without worrying about conflicting with an Internet network. However, when these networks are connected to the Internet, they must translate their private IP network addresses into public IP addresses (NAT) in order to be routable. By doing this, a large number of hosts behind a firewall can take turns or share a few public addresses when accessing the Internet.

Address	Mask	Range
10.0.0.0	255.0.0.0	10.0.0.0–10.255.255.255
172.16.0.0	255.240.0.0	172.16.0.0–172.31.255.255
192.168.0.0	255.255.0.0	192.168.0.0–192.168.255.255

Table 15-1 Private Addresses Specified in RFC 1918

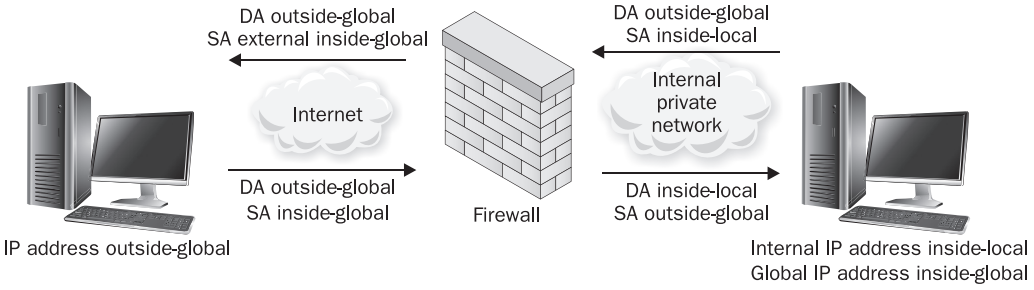

Figure 15-1 Network Address Translation

NAT is usually implemented in a firewall separately from the policy or rule set. It is useful to remember that just because a NAT has been defined to translate addresses between one host and another, it does not mean those hosts will be able to communicate. This is controlled by the policy defined in the firewall rule set.

When hosts have both public and private IP addresses, the IP information contained within a packet header will change depending on where the packet is viewed. For the purposes of this discussion, the addresses when viewed on the trusted side of the firewall will be referred to as *local addresses*. Once the packet crosses the firewall and is translated, the addresses will be called the host's *global addresses*. These terms, as depicted in Figure 15-1, will be used in the following sections to describe the various types and nuances of NAT. In this figure and the other figures in this chapter, the abbreviations "DA" and "SA" refer to "destination address" and "source address" respectively.

Static NAT

A static NAT configuration always results in the same address translation. The host is defined with one local address and a corresponding global address in a 1:1 relationship, and they don't change. The static NAT translation rewrites the source and destination IP addresses as required for each packet as it travels through the firewall. No other part of the packet is affected. This is typically used for internal servers that need to be reachable from the Internet reliably on an IP address that doesn't change. See Figure 15-2.

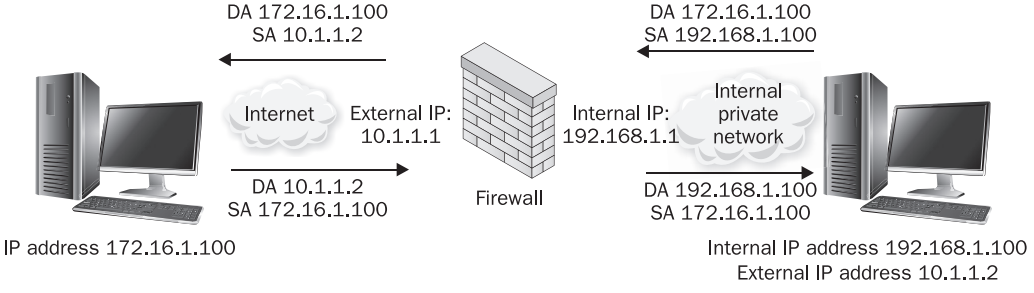

Figure 15-2 NAT replacing global terms with actual IP addresses

Because of this simplistic approach, most protocols will be able to traverse a static NAT without problems. The most common use of static NAT is to provide Internet access to a trusted host inside the firewall perimeter, or inbound access to a specific host, such as a web server that needs to be accessible via a public IP address.

Dynamic NAT

Dynamic NAT is used to map a group of inside local addresses to one or more global addresses. The global address set is usually smaller than the number of inside local addresses, and the conservation of addresses intended by RFC 1918 is accomplished by overlapping this address space. Dynamic NAT is usually implemented by simply creating static NATs when an inside host sends a packet through the firewall. The NAT is then maintained in the firewall tables until some event causes it to be terminated. This event is often a timer that expires after a predefined amount of inactivity from the inside host, thus removing the NAT entry. This address can then be reused by a different host.

One advantage of dynamic NAT over static NAT is that it provides a constantly changing set of IP addresses from the perspective of an Internet-based attacker, which makes targeting individual systems difficult. The greatest disadvantage of dynamic NAT is the limit on the number of concurrent users on the inside who can access external resources simultaneously. The firewall will simply run out of global addresses and not be able to assign new ones until the idle timers start freeing up global addresses.

Port Address Translation

With Port Address Translation (PAT), the entire inside local address space can be mapped to a single global address. This is done by modifying the communication port addresses in addition to the source and destination IP addresses. Thus, the firewall can use a single IP address for multiple communications by tracking which ports are associated with which sessions. In the example depicted in Figure 15-3, the sending host initiates a web connection on source port 1045. When the packet traverses the firewall, in addition to replacing the source IP address, the firewall translates the source port to port 5500 and creates an entry in a mapping table for use in translating future packets. When the firewall receives a packet back for destination port 5500, it will know how to translate the response properly. Using this system, thousands of sessions can be PATed behind a single IP address simultaneously.

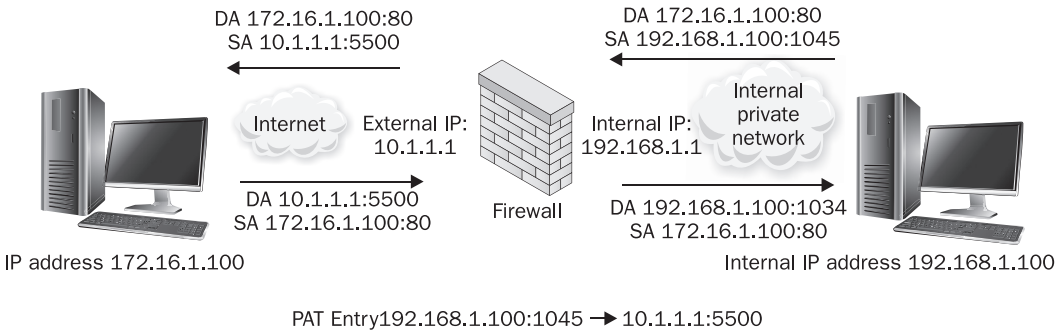

Figure 15-3 An example of Port Address Translation

PAT provides an increased level of security because it cannot be used for incoming connections. However, a downside to PAT is that it limits connection-oriented protocols, such as TCP.

Some firewalls will try to map UDP and ICMP connections, allowing DNS, Network Time Protocol (NTP), and ICMP echo replies to return to the proper host on the inside network. However, even those firewalls that do use PAT on UDP cannot handle all cases. With no defined end of session, they will usually time out the PAT entry after some predetermined time. This timeout period must be set to be relatively short (from seconds to a few minutes) to avoid filling the PAT table (although, on modern firewalls, the tables used for these sessions, commonly called *translation tables*, can frequently handle tens of thousands or even millions of sessions).

Connection-oriented protocols have a defined end of session built into them that can be picked up by the firewall. The timeout period associated with these protocols can be set to a relatively long period (hours or even days).

Auditing and Logging

Firewalls are excellent auditors. Given plenty of disk space or remote logging capabilities, they can record any traffic that passes through them. Attack attempts will leave evidence in logs, and if administrators are watching systems diligently, attacks can be detected before they are successful. Therefore, it is important that system activity be logged and monitored. Firewalls should record system events that are both successful and unsuccessful. Verbose logging and timely reviews of those logs can alert administrators to suspicious activity before a serious security breach occurs. Since this can generate a huge volume of log traffic, the logs are best sent to a Security Information and Event Management (SIEM) system that can filter, analyze, and perform heuristic behavior detection to help the network and security administrators.

Additional Firewall Capabilities

Modern firewalls can do more than manage application communications and behaviors; they can also assist in other areas of network quality and performance. Features vary by manufacturer and brand, but you will probably find that you can solve other problems in your environment with the same firewall you use to secure network traffic.

Application and Website Malware Execution Blocking

In the old days (just a few years ago), viruses required a user to click on some disguised link or button to execute. If the end users were sophisticated enough to recognize the virus writers' tricks, these viruses wouldn't get very far. Modern malware can execute and spread itself without the intervention of end users. Through automatic, browser-based execution of code (via ActiveX or Java, for example), simply opening a web page can activate a virus. Adobe PDF files can also transmit malware, due to their extensive underlying application framework. Firewalls with advanced anti-malware capability should be able to detect these "invisible" malware vectors and stop them in their tracks. They should also be able to block the communication "back home" to a command and control (CnC) server once malware

successfully implants itself on a victim system and tries to reach back to its controller for instructions.

Antivirus

Firewalls that are sophisticated enough to detect malware can (and should) block it on the network. Worms that try to propagate and spread themselves automatically on the network, and malware that tries to "phone home," can be stopped by the firewall, confining their reach. Malware control solutions should be layered, and the firewall can form an important component of a network-based malware blocking capability to complement your organization's endpoint antivirus software.

Intrusion Detection and Intrusion Prevention

Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are discussed in more detail in Chapter 18. Firewalls can provide IDS and IPS capabilities at the network perimeter, which can be a useful addition or substitution for standard purpose-built intrusion detection and prevention systems, especially in a layered strategy.

Web Content (URL) Filtering and Caching

The firewall is optimally positioned on the network to filter access to web sites (between an organization's internal networks and the Internet). You can choose to implement a separate URL filtering system or service, or you can get a firewall that has the capability built-in. Today's firewalls are demonstrating web content filtering capabilities that rival those of purpose-built systems, so you may be able to save money by doing the filtering on the firewall—especially if it doesn't cost extra.

E-Mail (Spam) Filtering

As with web content filtering, modern firewalls can subtract the spam from your e-mail messages before they get delivered to your mail server. You can sign up for an external service or buy a purpose-built spam filter instead, but with a firewall that includes this capability, you have another option.

Enhance Network Performance

Firewalls need to be able to run at "wire speed"—fast enough to avoid bottlenecking application traffic. They should be able to perform all the functions that have been enabled without impacting performance. In addition, firewalls should be able to allocate network bandwidth to the most critical applications to ensure QoS, without sacrificing filtering functionality. As firewall features continue to become more sophisticated, the underlying hardware needs to keep up. If your network has a low tolerance for performance impact, you'll want to consider firewall platforms that are built for speed.

Firewall Design

Firewalls may be software based or, more commonly, purpose-built appliances. Sometimes the firewalling functions are actually provided by a collection of several different devices. The specific features of the firewall platform and the design of the network where the firewall lives

are key components of securing a network. To be effective, firewalls must be placed in the right locations on the network, and configured effectively. Best practices include

- All communications must pass through the firewall. The effectiveness of the firewall is greatly reduced if an alternative network routing path is available; unauthorized traffic can be sent through a different network path, bypassing the control of the firewall. Think of the firewall in terms of a lock on your front door. It can be the best lock in the world, but if the back door is unlocked, intruders don't have to break the lock on the front door—they can go around it. The door lock is relied upon to prevent unauthorized access through the door, and a firewall is similarly relied upon to prevent access to your network.
- The firewall permits only traffic that is authorized. If the firewall cannot be relied upon to differentiate between authorized and unauthorized traffic, or if it is configured to permit dangerous or unneeded communications, its usefulness is also diminished.
- In a failure or overload situation, a firewall must always fail into a "deny" or closed state, under the principle that it is better to interrupt communications than to leave systems unprotected.
- The firewall must be designed and configured to withstand attacks upon itself. Because the firewall is relied upon to stop attacks, and nothing else is deployed to protect the firewall itself against such attacks, it must be hardened and capable of withstanding attacks directly upon itself.

Firewall Strengths and Weaknesses

A firewall is just one component of an overall security architecture. Its strengths and weaknesses should be taken into consideration when designing network security.

Firewall Strengths

Consider the following firewall strengths when designing network security:

- Firewalls are excellent at enforcing security policies. They should be configured to restrict communications to what management has determined and agreed with the business to be acceptable.
- Firewalls are used to restrict access to specific services.
- Firewalls are transparent on the network—no software is needed on end-user workstations.
- Firewalls can provide auditing. Given plenty of disk space or remote logging capabilities, they can log interesting traffic that passes through them.
- Firewalls can alert appropriate people of specified events.

Firewall Weaknesses

You must also consider the following firewall weaknesses when designing network security:

- Firewalls are only as effective as the rules they are configured to enforce. An overly permissive rule set will diminish the effectiveness of the firewall.
- Firewalls cannot stop social engineering attacks or an authorized user intentionally using their access for malicious purposes.
- Firewalls cannot enforce security policies that are absent or undefined.
- Firewalls cannot stop attacks if the traffic does not pass through them.

Firewall Placement

A firewall is usually located at the network perimeter, directly between the network and any external connections. However, additional firewall systems can be located inside the network perimeter to provide more specific protection to particular hosts with higher security requirements. The placement of firewalls in a network and overall security design was discussed in greater detail in Chapter 13.

Firewall Configuration

When building a rule set on a firewall, consider the following practices:

- Build rules from most to least specific. Most firewalls process their rule sets from top to bottom and stop processing once a match is made. Putting more specific rules on top prevents a general rule from hiding a specific rule further down the rule set.
- Place the most active rules near the top of the rule set. Screening packets is a processor-intensive operation, and as mentioned earlier, a firewall will stop processing the packet after matching it to a rule. Placing your popular rules first or second, instead of 30th or 31st, will save the processor from going through over 30 rules for every packet. In situations where millions of packets are being processed and rule sets can be thousands of entries in length, CPU savings could be considerable.
- Configure all firewalls to drop "impossible" or "unroutable" packets from the Internet such as those from an outside interface with source addresses matching the internal network, RFC 1918 "private" IP addresses, and broadcast packets. None of these would be expected from the Internet, so if they are seen, they represent unwanted traffic such as that produced by attackers.

Summary

This chapter provided an in-depth overview of firewalls, their relevance to applications and OSI layer seven, and their roles in protecting the network. Good security practices dictate that firewalls should be deployed between any two networks of differing security requirements; this includes perimeter connections, as well as connections between sensitive internal networks.

References

Becher, Michael. *Web Application Firewalls: Applied Web Application Security*. AV Akademikerverlag, 2012.

Liu, Alex. *Firewall Design and Analysis*. World Scientific Publishing Company, 2010.

Miller, Lawrence. *Next Generation Firewalls for Dummies*. Wiley, 2011.

Stewart, J. Michael. *Network Security, Firewalls, and VPNs*. Jones & Bartlett Learning, 2010.

Strassberg, Keith, Richard Gondek, and Gary Rollie. *Firewalls: The Complete Reference*. McGraw-Hill, 2002.

Whitman, Michael, Herbert Mattord, and Andrew Green. *Guide to Firewalls and VPNs*. 3rd ed. Delmar Cengage Learning, 2011.