Chapter 7

Locking Down Your XenApp Server

Solutions in this chapter:

- Protecting Your Server (and Its Parts)
- Protecting Your Data
- Planning Physical Security
- Security Measures and Objectives

☑ Summary

☑ Solutions Fast Track

☑ Frequently Asked Questions

Introduction

To protect assets from risks that were identified as possible threats to a business, countermeasures must be implemented. Servers will need certain configurations to provide security, and plans must be put into practice. Compare the risks faced by an organization with an operating system's features to find support that will address certain threats. Configuring the server to use these services or tools can assist in dealing with potential problems. For example, installing AD and using domain controllers on a network can heighten security and provide the ability to control user access and security across the network. In the same way, configuring a file server to use EFS so that data on the server's hard disk is encrypted can augment file security. Using security features in an operating system allows you to minimize many potential threats.

Protecting Your Server (and Its Parts)

As we have said many times in this book that your job as a XenApp administrator is to not only make applications and resources available to your users, but to do so securely while still maintaining a productive environment that is usable. There are many tools and resources available to assist you in completing the task of protecting your server. However, sometimes the most obvious things are often overlooked. For example, let's say that an industrious employee can quickly reboot one of your XenApp servers and change the BIOS configuration to boot from a USB device that they connect to the server, or even a CD that has "questionable" content on it. Sound bizarre? Perhaps, but how long would that employee need to compromise the system if he were to have physical access to one of your servers? What could he do? Your job as security administrators is to think about situations just like this one, and in this section, we will cover some of the ways you can protect your server (and its parts.)

System BIOS Lockdown

You can adopt the best security policies and implement the latest software and you can still have a security incident. Without adequately ensuring the physical integrity of your servers, you could be easily compromised by an insider, like coworker, contractor, visitor or even a friend. Therefore, you must ensure that even the most mundane ways of gaining access to your systems is thoroughly considered.

All your security measures are useless if an attacker can gain physical access to your system. What would happen if an attacker were to gain access to one of your servers, could he then take a bootable CD and reboot your server and have it boot from the CD because your BIOS setting allows for that – or worse yet, your BIOS is not protected.

To prevent this, use the system BIOS to disable boot devices other than the hard disk (or, if that's not possible, select the hard disk as the first boot device). For computers located in hard-to-protect public areas, consider removing floppy and CD/DVD drives, and disabling or removing USB and FireWire ports, to prevent people from booting the PC with a Linux disc, IPod or flash memory USB drive.

Password protect your system BIOS. Most types of BIOS let you create a user password that must be entered to permit the system to start up. If your BIOS supports it, an administrative password will prevent attackers from changing BIOS settings (including the boot password).

You should have this password stored in a secure location in case you need to make changes to the system later. Keep in mind that just because you password protect your system BIOS you will thwart an attackers efforts. Some systems accept *master* passwords, lists of which appear on the Internet. By entering particular key combinations an attacker may be able to sidestep BIOS password security on certain systems.

Additionally, anyone with the opportunity to open the system's case can clear the passwords by moving a jumper on the motherboard, or by disconnecting the battery that powers the BIOS settings' memory chip. If you're worried about that happening, get a lock for the case itself and have the server in a locked server cabinet.

USB Blockers

USB blockers can be installed to minimize the security risks imposed by using portable storage media and portable devices in your enterprise, such as:

- Memory sticks
- Jump drive
- Personal Digital Assistants (PDAs)
- Cell Phones
- MP3 Players
- Digital Cameras
- Scanners
- Hard disk drives
- CD/DVD
- Printers

By using these devices, data can be retrieved and often changed while bypassing established security guidelines. Of course, this frequently causes irritation to administrators. Because of that, enterprises are threatened not only of the danger of the data theft and integrity compromise, but also of the intentional or unintentional "import" of viruses, Trojan horses and other damaging software.

Most USB blockers are built up on the Windows Device Manager and can be configured to prevent new or unknown devices from being accessible. These blockers are typically installed as a software component within the Windows operating system and can then be configured via Windows Group Policy.

Alarms

You can utilize XenApp Resource Manager with or without the summary database to establish a means of producing alarms based on metrics you configure and want to monitor.

The first step is to establish the metrics that you want to track. This process is easier than establishing the thresholds for those metrics; in short, this initial step is more art than science.

However, you can leverage Resource Manager and/or Performance Monitor in the creation of this baseline to allow you to more accurately establish the metrics and their thresholds that you will track.

Each metric has both yellow and red thresholds as configured on the right. Yellow represents the warning state, meaning something is slightly out of ordinary but not considered catastrophic yet. Red is reserved for those metrics that indicate a severe state that will need immediate attention. As we have stated several times, choosing the correct values to enter into the Threshold Configuration is definitely a challenge that we all face. The reality is that no two Presentation Server environments are identical; every environment has its own specific server hardware, operating systems, applications, and so on. Add to that the nuisance of differences in patch levels, software configuration, and usage patterns within various applications and you can quickly see why this will present a "tweaking" exercise for your environment. As a general rule of thumb, however, you can use the Visual Threshold Configuration window to get good idea of what your current environment looks like so you can make a more educated guess as to the values to enter.

Configuring alerts is a two-step process. Step one is to configure the individual metrics you want to be alerted on and the method to be alerted with. The second step is to enable alerting and configure the alerting methods. Alerts can be sent to e-mails, SNMP traps, or Short Message Service (SMS) pages. The Alerts tab of a given metric's properties allows for the configuration of these options. Basically, enable the way you want to be alerted when that particular metric enters an alert state, such as the transition to red. As a best practice, it is considered good etiquette to notify that the alert state is over by sending confirmation of the transition back to a green state.

Intrusion Detection Systems

"Danger! Will Robinson! Intruder Alert!" When we heard that ominous announcement emanating from a robot as it twisted and turned with arms thrashing and head spinning, we sat galvanized to our televisions waiting for the intruder to reveal itself. Would this be the end of Will Robinson as we knew him?

All right, this might be a bit dramatic for a prelude to a discussion of intrusion detection, but with most security administrators, when a beeper goes off there is a moment of anxiety. Is this the big one? Did they get in? Do they own my network? Do they own my data?

These and many other questions flood the mind of the well-prepared security administrator. Conversely, the ill-prepared security administrator, being totally unaware of the intrusion, experiences little anxiety. For him, the anxiety comes later.

Okay, so how can a security-minded administrator protect his network from intrusions? The answer to that question is quite simple, with an intrusion detection system.

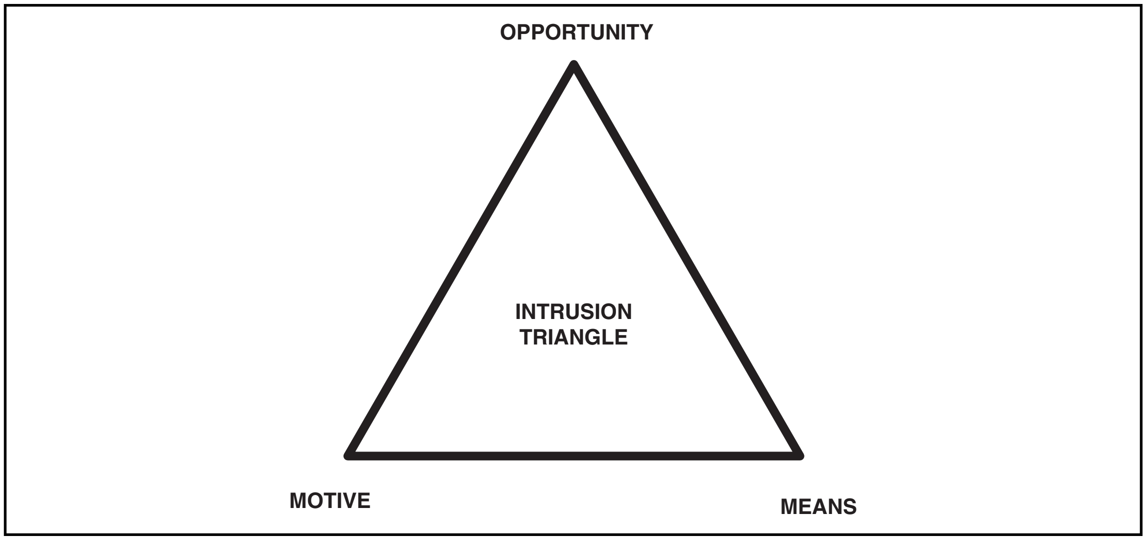
NOTE

Intrusion detection works in conjunction with firewalls in various ways. One of the ways is to use intrusion detection is to test your firewall rules to make sure they are working properly. One of the other ways is to use intrusion detection and firewalls to set rules for a firewall.

What Is an Intrusion?

Borrowing from the law enforcement community, crime prevention specialists use a model called the "Crime Triangle" to explain that certain criteria must exist before a crime can occur. We can adapt this same triangle to network security: the same three criteria must exist before a network security breach can take place. The three "legs" or points of the triangle are shown in Figure 7.1.

Figure 7.1 All Three Legs of the Triangle Must Exist for a Network Intrusion to Occur



Let's look at each point individually:

- **Motive** An intruder must have a reason to want to breach the security of your network (even if the reason is "just for fun"); otherwise, he/she won't bother.
- **Means** An intruder must have the ability (either the programming knowledge, or, in the case of "script kiddies," the intrusion software written by others), or he/she won't be able to breach your security.
- **Opportunity** An intruder must have the chance to enter the network, either because of flaws in your security plan, holes in a software program that open an avenue of access, or physical proximity to network components; if there is no opportunity to intrude, the would-be hacker will go elsewhere.

If you think about the three-point intrusion criteria for a moment, you'll see that there is really only one leg of the triangle over which you, as the network administrator or security specialist, have any control. It is unlikely that you can do much to remove the intruder's motive. The motive is likely to be built into the type of data you have on the network or even the personality of the intruder him/herself. It is also not possible for you to prevent the intruder from having or obtaining the means to breach your security. Programming knowledge is freely available, and there are many experienced hackers out there who are more than happy to help out less sophisticated ones. The one thing that you can affect is the opportunity afforded the hacker.

What Is Intrusion Detection?

Webster's dictionary defines an intrusion as "the act of thrusting in, or of entering into a place or state without invitation, right, or welcome." When we speak of intrusion detection, we are referring to the act of detecting an unauthorized intrusion by a *computer* on a *network*. This unauthorized access, or intrusion, is an attempt to compromise, or otherwise do harm, to other network devices.

An intrusion detection system (IDS) is the high-tech equivalent of a burglar alarm—a burglar alarm configured to monitor access points, hostile activities, and known intruders. The simplest way to define an IDS might be to describe it as a specialized tool that knows how to read and interpret the contents of log files from routers, firewalls, servers, and other network devices. Furthermore, an IDS often stores a database of known attack signatures and can compare patterns of activity, traffic, or behavior it sees in the logs it is monitoring against those signatures to recognize when a close match between a signature and current or recent behavior occurs. At that point, the IDS can issue alarms or alerts, take various kinds of automatic action ranging from shutting down Internet links or specific servers to launching backtraces, and make other active attempts to identify attackers and actively collect evidence of their nefarious activities.

By analogy, an IDS does for a network what an antivirus software package does for files that enter a system: It inspects the contents of network traffic to look for and deflect possible attacks, just as an antivirus software package inspects the contents of incoming files, e-mail attachments, active Web content, and so forth to look for virus signatures (patterns that match known malware) or for possible malicious actions (patterns of behavior that are at least suspicious, if not downright unacceptable).

To be more specific, intrusion detection means detecting unauthorized use of or attacks on a system or network. An IDS is designed and used to detect and then to deflect or deter (if possible) such attacks or unauthorized use of systems, networks, and related resources. Like firewalls, IDSes can be software based or can combine hardware and software (in the form of preinstalled and preconfigured stand-alone IDS devices). Often, IDS software runs on the same devices or servers where firewalls, proxies, or other boundary services operate; an IDS *not* running on the same device or server where the firewall or other services are installed will monitor those devices closely and carefully. Although such devices tend to operate at network peripheries, IDSes can detect and deal with insider attacks as well as external attacks.

IDSes vary according to a number of criteria. By explaining those criteria, we can explain what kinds of IDSes you are likely to encounter and how they do their jobs. First and foremost, it is possible to distinguish IDSes by the kinds of activities, traffic, transactions, or systems they monitor. IDSes can be divided into network-based, host-based, and distributed. IDSes that monitor network backbones and look for attack signatures are called *network-based IDSes*, whereas those that operate on hosts defend and monitor the operating and file systems for signs of intrusion and are called *host-based IDSes.* Groups of IDSes functioning as remote sensors and reporting to a central management station are known as Distributed IDS (DIDS).

In practice, most commercial environments use some combination of network, and host, and/or application-based IDS systems to observe what is happening on the network while also monitoring key hosts and applications more closely. IDSes can also be distinguished by their differing approaches to event analysis. Some IDSes primarily use a technique called *signature detection*. This resembles the way many antivirus programs use virus signatures to recognize and block infected files, programs, or active Web content from entering a computer system, except that it uses a database of traffic or activity patterns related to known attacks, called *attack signatures*. Indeed, signature detection is the most widely used approach in commercial IDS technology today. Another approach is called *anomaly detection*. It uses rules or predefined concepts about "normal" and "abnormal" system activity (called *heuristics*) to distinguish anomalies from normal system behavior and to monitor, report on, or block anomalies as they occur. Some anomaly detection IDSes implement user profiles. These profiles are baselines of normal activity and can be constructed using statistical sampling, rule-base approach or neural networks.

Literally hundreds of vendors offer various forms of commercial IDS implementations. Most effective solutions combine network- and host-based IDS implementations. Likewise, the majority of implementations are primarily signature based, with only limited anomaly-based detection capabilities present in certain specific products or solutions. Finally, most modern IDSes include some limited automatic response capabilities, but these usually concentrate on automated traffic filtering, blocking, or disconnects as a last resort. Although some systems claim to be able to launch counterstrikes against attacks, best practices indicate that automated identification and backtrace facilities are the most useful aspects that such facilities provide and are therefore those most likely to be used.

IDSes are classified by their functionality and are loosely grouped into the following three main categories:

- Network-Based Intrusion Detection System (NIDS)
- Host-Based Intrusion Detection System (HIDS)
- Distributed Intrusion Detection System (DIDS)

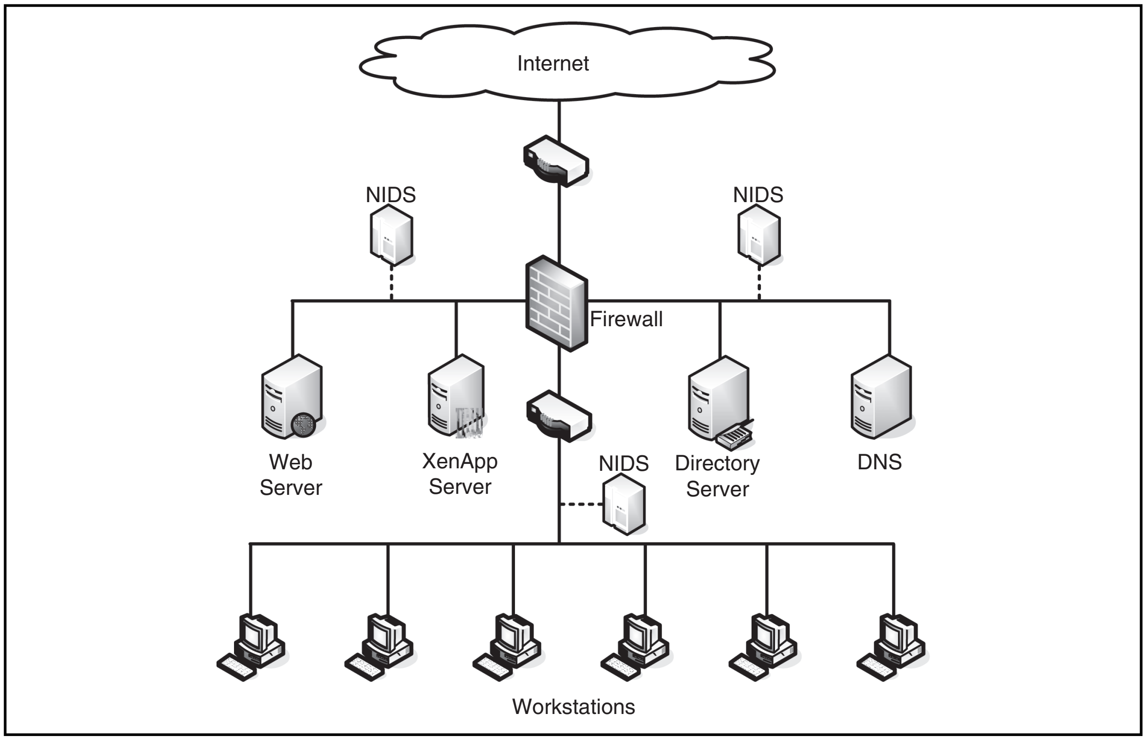
Network IDS

The NIDS derives its name from the fact that it monitors the entire network. More accurately, it monitors an entire network segment. Normally, a computer network interface card (NIC) operates in nonpromiscuous mode. In this mode of operation, only packets destined for the NICs specific media access control (MAC) address are forwarded up the stack for analysis. The NIDS must operate in promiscuous mode to monitor network traffic not destined for its own MAC address. In promiscuous mode, the NIDS can eavesdrop on all communications on the network segment. Operation in promiscuous mode is necessary to protect your network. However, in view of emerging privacy regulations, monitoring network communications is a responsibility that must be considered carefully.

In Figure 7.2, we see a network using three NIDS. The units have been placed on strategic network segments and can monitor network traffic for all devices on the segment. This configuration represents a standard perimeter security network topology where the screened subnets on the DMZ housing the public servers are protected by NIDS. When a public server is compromised on a screened subnet, the server can become a launching platform for additional exploits. Careful monitoring is necessary to prevent further damage.

The internal host systems inside the firewall are protected by an additional NIDS to mitigate exposure to internal compromise. The use of multiple NIDS within a network is an example of a defense-in-depth security architecture.

Figure 7.2 NIDS Network



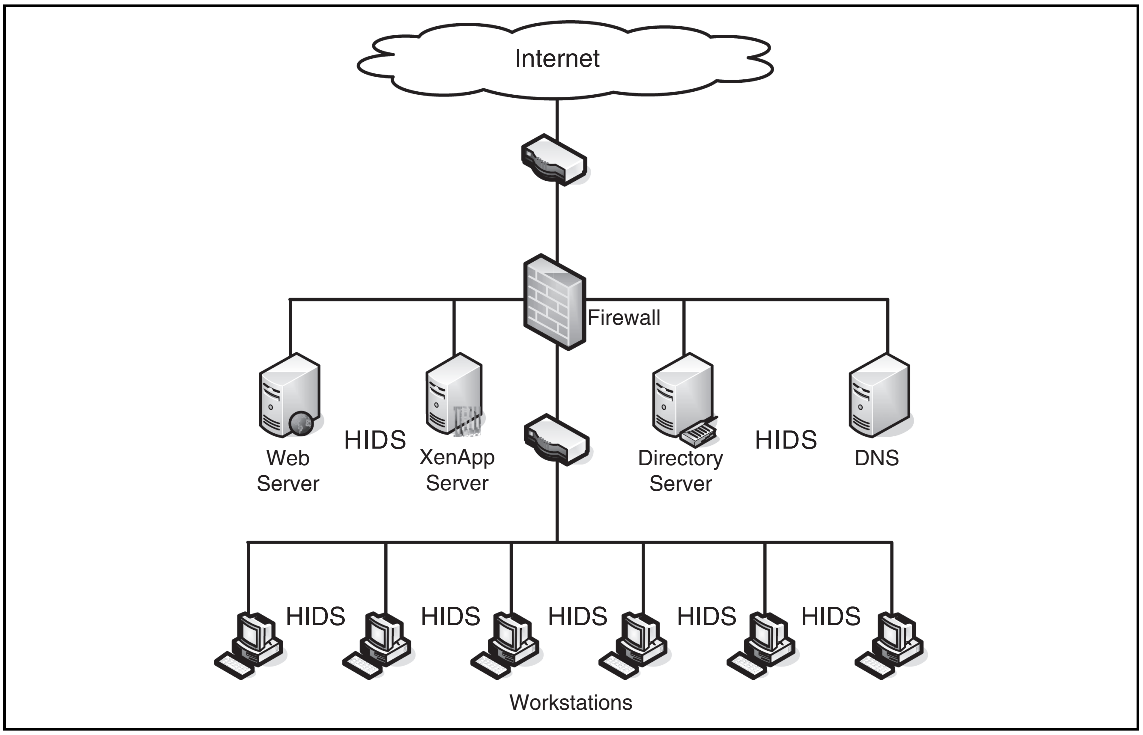
Host-Based IDS

HIDS differ from NIDS in two ways. HIDS protects only the host system on which it resides, and its network card operates in nonpromiscuous mode. Nonpromiscuous mode of operation can be an advantage in some cases, because not all NICs are capable of promiscuous mode. In addition, promiscuous mode can be CPU intensive for a slow host machine. HIDS can be run directly on the firewall as well, to help keep the firewall secure.

Another advantage of HIDS is the ability to tailor the ruleset to a specific need. For example, there is no need to interrogate multiple rules designed to detect DNS exploits on a host that is not running Domain Name Services. Consequently, the reduction in the number of pertinent rules enhances performance and reduces processor overhead.

Figure 7.3 depicts a network using HIDS on specific servers and host computers. As previously mentioned, the ruleset for the HIDS on the mail server is customized to protect it from mail server exploits, while the Web server rules are tailored for Web exploits. During installation, individual host machines can be configured with a common set of rules. New rules can be loaded periodically to account for new vulnerabilities.

Figure 7.3 HIDS Network



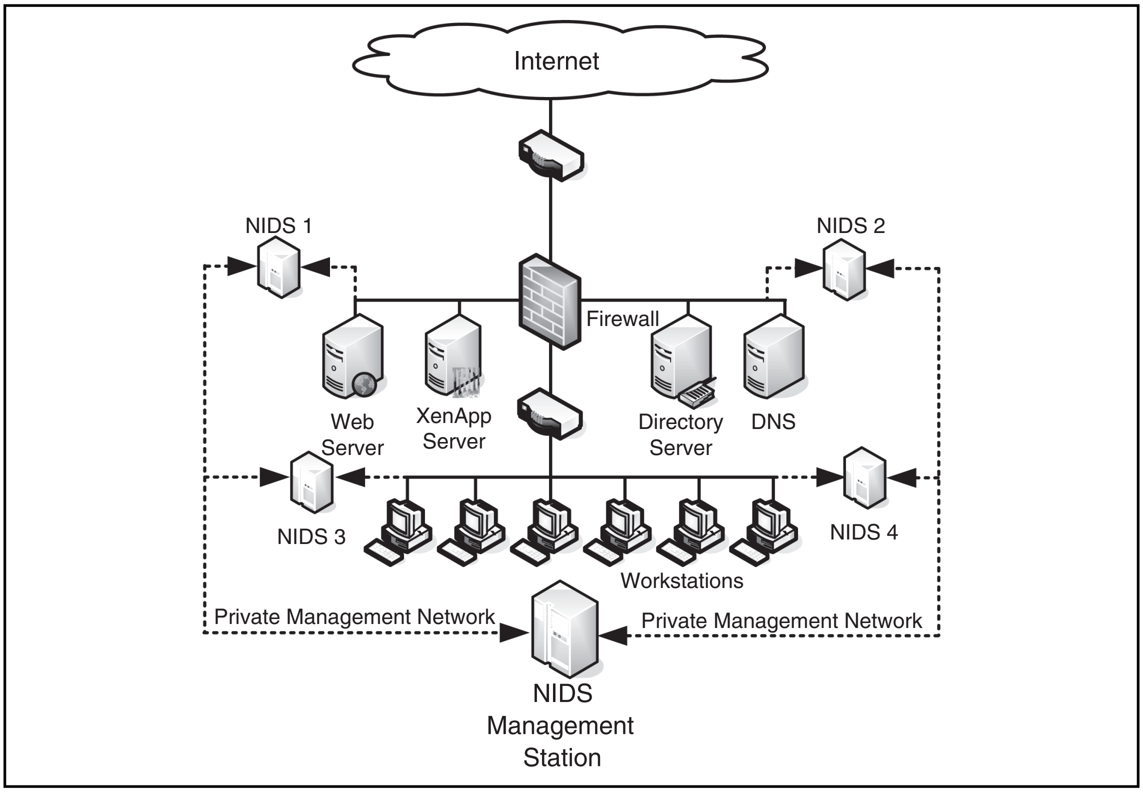
Distributed IDS

The standard DIDS functions in a Manager/Probe architecture. NIDS detection sensors are remotely located and report to a centralized management station. Attack logs are periodically uploaded to the management station and can be stored in a central database; new attack signatures can be downloaded to the sensors on an as-needed basis. The rules for each sensor can be tailored to meet its individual needs. Alerts can be forwarded to a messaging system located on the management station and used to notify the IDS administrator.

In Figure 7.4, we see a DIDS system comprised of four sensors and a centralized management station. Sensors NIDS 1 and NIDS 2 are operating in stealth promiscuous mode and are protecting the public servers. Sensors NIDS 3 and NIDS 4 are protecting the host systems in the trusted computing base. The DIDS are on the outside of the firewall, usually on the DMZ or outside.

The network transactions between sensor and manager can be on a private network, as depicted, or the network traffic can use the existing infrastructure. When using the existing network for management data, the additional security afforded by encryption, or VPN technology, is highly recommended.

Figure 7.4 DIDS Network



In a DIDS, complexity abounds. The scope and functionality varies greatly from manufacturer to manufacturer, and the definition blurs accordingly. In a DIDS, the individual sensors can be NIDS, HIDS, or a combination of both. The sensor can function in promiscuous mode or nonpromiscuous mode. However, in all cases, the DIDS' single defining feature requires that the distributed sensors report to a centralized management station.

Why Are Intrusion Detection Systems Important?

Everyone is familiar with the oft-used saying, "What you don't know can't hurt you." However, anyone who has ever bought a used automobile has learned, first hand, the absurdity of this statement. In the world of network security, the ability to know when an intruder is engaged in reconnaissance, or other malicious activity, can mean the difference between being compromised and not being compromised. In addition, in some environments, what you don't know can directly affect employment—yours.

IDSes can detect ICMP and other types of network reconnaissance scans that might indicate an impending attack. In addition, the IDS can alert the admin of a successful compromise, which allows him the opportunity to implement mitigating actions before further damage is caused.

IDSes provide the security administrator with a window into the inner workings of the network, analogous to an x-ray or a blood test in the medical field. The ability to analyze the internal network traffic and to determine the existence of network viruses and worms is not altogether different from techniques used by the medical profession. The similarity of network viruses and worms to their biological counterparts has resulted in their medical monikers. IDSes provide the microscope necessary to detect these invaders. Without the aid of intrusion detection, a security administrator is vulnerable to exploits and will become aware of the presence of exploits only after a system crashes or a database is corrupted.

Why Are Attackers Interested in Me?

"The Attack of the Zombies"—sounds a lot like an old B-grade movie, doesn't it? Unfortunately, in this case, it is not cinema magic. Zombie attacks are real and cost corporations and consumers billions. Zombies are computerized soldiers under the control of nefarious hackers, and in the process of performing distributed denial-of-service (DDoS) attacks, they blindly carry out the will of their masters.

In February 2000, a major DDoS attack blocked access to eBay, Amazon.com, AOL-TimeWarner, CNN, Dell Computers, Excite, Yahoo!, and other e-commerce giants. The damage done by this DDoS ranged from slowdown to complete system outages. The U.S. Attorney General instructed the FBI to launch a criminal investigation. This historical attack was perpetrated by a large group of compromised computers operating in concert.

The lesson to be learned from this event is that no network is too small to be left unprotected. If a hacker can use your computer, he will. The main purpose of the CodeRed exploit was to perform a DDoS on the White House Web site. It failed, due only to the author's oversight in using a hard-coded IP address instead of Domain Name Services. The exploit compromised over a million computers, ranging from corporate networks to home users.

In light of the recent virus activity, the growth of the information security industry, and taking into account government-sponsored hacking, the use of an IDS such can prove crucial in the protection of the world's network infrastructure.

Where Does an IDS Fit with the Rest of My Security Plan?

IDSes are a great addition to a network's defense-in-depth architecture. They can be used to identify vulnerabilities and weaknesses in your perimeter protection devices; for example, firewalls and routers. The firewall rules and router access lists can be verified regularly for functionality. In the event these devices are reconfigured, the IDS can provide auditing for change management control.

IDS logs can be used to enforce security policy and are a great source of forensic evidence. Inline IDSes can halt active attacks on your network while alerting administrators to their presence.

Properly placed IDSes can alert you to the presence of internal attacks. Industry analysis of percentages varies. However, the consensus is that the majority of attacks occur from within.

An IDS can detect failed administrator login attempts and recognize password-guessing programs. Configured with the proper ruleset, it can monitor critical application access and immediately notify the system administrator of possible breaches in security.

Doesn't My Firewall Serve as an IDS?

At this point, you may hazard the question, "doesn't my firewall serve as an IDS?" Absolutely Not! Having said that, we shall try to stop the deluge of scorn from firewall administrators who might take exception to the statement. Admittedly, a firewall can be configured to detect certain types of intrusions, such as an attempt to access the Trojan backdoor SubSeven's port 27374. In addition, it could be configured to generate an alert for any attempt to penetrate your network. In the strictest sense this would be an IDS function.

However, it is asking enough of the technology to simply determine what should and shouldn't be allowed into or out of your network without expecting it to analyze the internal contents of every packet. Even a proxy firewall is not designed to examine the contents of all packets; the function would be enormously CPU intensive. Nevertheless, a firewall should be an integral part of your defense-in-depth, with its main function being a gatekeeper and a filter (see Table 7.1).

Table 7.1 Comparing Firewalls and IDS

Functionality	Firewall	IDS
Detects unauthorized and malicious access by a computer	Yes	
	Yes	
Uses signatures to identify malicious intrusions	No	Yes
Defines borders on a trusted network from an untrusted network	Yes	No
Enforces Network Security Policies	Yes	Yes
Can detect failed administrator login attempts and recognize password-guessing programs	No	Yes
Used to identify vulnerabilities and weaknesses in your perimeter protection	No	Yes
Defines network traffic flow	Yes	No
Detects Trojan horses and Backdoors	No	Yes

Firewalls and IDS do both enforce network policy, but how they implement it is completely different. An IDS is a reconnaissance system: It collects information and will notify you of what it's found. An IDS can find any type of packet it's designed to find by a defined signature.

A firewall, on the other hand, is a like a dragon protecting the castle. It keeps out the untrusted network traffic, and only allows in what it has defined as being acceptable. For example, if an attacker has managed to compromise a Web server and uses it to store contraband (for example, pornographic materials, pirated software), your firewall will not detect this. However, if your Web server is being used for inappropriate content, this can be discovered through your IDS.

Both firewall logs and IDS logs can provide you with information to help with computer forensics or any incident handling efforts. If a system is compromised, you will have some logs on what has been going on—through both the firewall and the IDS.

What makes an IDS necessary for a defense in depth is that it can be used to identify vulnerabilities and weaknesses in your perimeter protection devices; in other words, firewalls and routers. Firewall rules and router access lists can be verified regularly for functionality. You can set up various IDS signatures to test your firewall to make sure it's not letting some undesired network traffic through the filter.

Where Else Should I Be Looking for Intrusions?

When computers that have been otherwise stable and functioning properly begin to perform erratically and periodically hang or show the Blue Screen of Death, a watchful security administrator should consider the possibility of a *buffer overflow attack*.

Buffer overflow attacks represent a large percentage of today's computer exploits. Failure of programmers to check input code has led to some of the most destructive and costly vulnerabilities to date.

Exploits that are designed to overflow buffers are usually operating system (OS) and application software specific. Without going into detail, the input to the application software is manipulated in such a manner as to cause a system error or "smash the stack" as it is referred to by some security professionals. At this point in the exploit, malicious code is inserted into the computer's process stack and the hacker gains control of the system.

In some cases, for the exploit to be successful, the payload, or malicious code, must access OS functions located at specific memory addresses. If the application is running on an OS other than that for which the exploit was designed, the results of overflowing the buffer will be simply a system crash and not a compromise; the system will appear to be unstable with frequent resets. Interestingly, in this situation the definition of the exploit changes from a system compromise to a DoS attack.

IDSes can alert you to buffer overflow attacks. Snort has a large arsenal of rules designed to detect these attacks; the following are just a few:

- Red Hat lprd overflow
- Linux samba overflow
- IMAP login overflow
- Linux mountd overflow

Using an IDS to Monitor My Company Policy

In today's litigious society, given the enormous legal interest in subjects such as downstream litigation and intellectual property rights, it would be prudent to consider monitoring for compliance with your company's security policy. Major motion picture companies have employed law firms specializing in Internet theft of intellectual property. Recently, many companies were sued because their employees illegally downloaded the motion picture *Spiderman*. Some of the employees involved were not aware that their computers were taking part in a crime. Nevertheless, the fines for damages were stiff—up to $100,000 in some cases.