# 7-Security Technology: Intrusion Detection and Prevention Systems, and other Tools

## Intrusion Detection and Prevention Systems

An **intrusion** occurs when an attacker attempts to gain entry into or disrupt the normal operations of an information system, almost always with the intent to do harm. Even when such attacks are self-propagating, as in the case of viruses and distributed denial-of-service attacks, they are almost always instigated by someone whose purpose is to harm an organization.
**Intrusion prevention** consists of activities that deter an intrusion. Some important intrusion prevention activities are writing and implementing good enterprise information security policy, planning and executing effective information security programs, installing and testing technology-based information security countermeasures (such as firewalls and intrusion detection systems), and conducting and measuring the effectiveness of employee training and awareness activities.
**Intrusion detection** consists of procedures and systems that identify system intrusions.
**Intrusion reaction** encompasses the actions an organization takes when an intrusion is detected.
**Intrusion correction** activities finalize the restoration of operations to a normal state and seek to identify the source and method of the intrusion in order to ensure that the same type of attack cannot occur again—thus reinitiating intrusion prevention.

Information security **intrusion detection systems** (**IDSs**) became commercially available in the late 1990s. An IDS works like a burglar alarm in that it detects a violation (some system activity analogous to an opened or broken window) and activates an alarm. This alarm can be audible and/or visual (producing noise and lights, respectively), or it can be silent (an e-mail message or pager alert). Many IDSs enable administrators to configure the systems to notify them directly of trouble via e-mail or pagers. The systems can also be configured—again like a burglar alarm—to notify an external security service organization of a "break-in."
A current extension of IDS technology is the intrusion prevention system (IPS), which can detect an intrusion and also prevent that intrusion from successfully attacking the organization by means of an active response. Because the two systems often coexist, the combined term intrusion detection and prevention system (IDPS) is generally used to describe current anti-intrusion technologies.

### IDPS Terminology

In order to understand IDPS operational behavior, you must first become familiar with some IDPS terminology.

- **Alert or alarm:** An indication that a system has just been attacked or is under attack. IDPS alerts and alarms take the form of audible signals, e-mail messages, pager notifications, or pop-up windows.
- **Evasion:** The process by which attackers change the format and/or timing of their activities to avoid being detected by the IDPS.
- **False attack stimulus:** An event that triggers an alarm when no actual attack is in progress. Scenarios that test the configuration of IDPSs may use false attack stimuli to determine if the IDPSs can distinguish between these stimuli and real attacks.
- **False negative:** The failure of an IDPS to react to an actual attack event. This is the most grievous failure, since the purpose of an IDPS is to detect and respond to attacks.
- **False positive:** An alert or alarm that occurs in the absence of an actual attack. A false positive can sometimes be produced when an IDPS mistakes normal system activity for an attack. False positives tend to make users insensitive to alarms and thus reduce their reactivity to actual intrusion events.
- **Site policy:** The rules and configuration guidelines governing the implementation and operation of IDPSs within the organization.
- **Site policy awareness:** An IDPS's ability to dynamically modify its configuration in response to environmental activity. A so-called smart IDPS can adapt its reactions in response to administrator guidance over time and circumstances of the current local environment. A smart IDPS logs events that fit a specific profile instead of minor events, such as file modification or failed user logins. The smart IDPS knows when it does not need to alert the administrator—for example, when an attack is using a known and documented exploit that the system is protected from.
- **True attack stimulus:** An event that triggers alarms and causes an IDPS to react as if a real attack is in progress. The event may be an actual attack, in which an attacker is at work on a system compromise

attempt, or it may be a drill, in which security personnel are using hacker tools to conduct tests of a network segment.

- **Tuning:** The process of adjusting an IDPS to maximize its efficiency in detecting true positives, while minimizing both false positives and false negatives.
- **Confidence value:** The measure of an IDPS's ability to correctly detect and identify certain types of attacks. The confidence value an organization places in the IDPS is based on experience and past performance measurements. The confidence value, which is based upon fuzzy logic, helps an administrator determine how likely it is that an IDPS alert or alarm indicates an actual attack in progress. For example, if a system deemed 90 percent capable of accurately reporting a denial-of-service attack sends a denial-of-service alert, there is a high probability that an actual attack is occurring.
- **Alarm filtering:** The process of classifying IDPS alerts so that they can be more effectively managed. An IDPS administrator can set up alarm filtering by running the system for a while to track what types of false positives it generates and then adjusting the alarm classifications. For example, the administrator may set the IDPS to discard alarms produced by false attack stimuli or normal network operations. Alarm filters are similar to packet filters in that they can filter items by their source or destination IP addresses, but they can also filter by operating systems, confidence values, alarm type, or alarm severity.
- **Alarm clustering and compaction:** A process of grouping almost identical alarms that happen at close to the same time into a single higher-level alarm. This consolidation reduces the number of alarms generated, thereby reducing administrative overhead, and also identifies a relationship among multiple alarms. This clustering may be based on combinations of frequency, similarity in attack signature, similarity in attack target, or other criteria that are defined by the system administrators.

## Why Use an IDPS?

According to the NIST documentation on industry best practices, there are several compelling reasons to acquire and use an IDPS:

1. To prevent problem behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system
2. To detect attacks and other security violations that are not prevented by other security measures
3. To detect and deal with the preambles to attacks
4. To document the existing threat to an organization
5. To act as quality control for security design and administration, especially in large and complex enterprises
6. To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors.

**Reasons**

One of the best reasons to install an IDPS is that they serve as deterrents by increasing the fear of detection among would-be attackers. If internal and external users know that an organization has an intrusion detection and prevention system, they are less likely to probe or attempt to compromise it, just as criminals are much less likely to break into a house that has an apparent burglar alarm.

Another reason to install an IDPS is to cover the organization when its network cannot protect itself against known vulnerabilities or is unable to respond to a rapidly changing threat environment. There are many factors that can delay or undermine an organization's ability to secure its systems from attack and subsequent loss. For example, even though popular information security technologies such as scanning tools (discussed later in this chapter) allow security administrators to evaluate the readiness of their systems, they may still fail to detect or correct a known deficiency or may perform the vulnerability-detection process too infrequently. In addition, even when a vulnerability is detected in a timely manner, it cannot always be corrected quickly.

IDPSs can also help administrators detect the preambles to attacks. Most attacks begin with an organized and thorough probing of the organization's network environment and its defenses. This initial estimation of the defensive state of an organization's networks and systems is called doorknob rattling and is accomplished by means of footprinting (activities that gather information about the organization and its network activities and

assets) and fingerprinting (activities that scan network locales for active systems and then identify the network services offered by the host systems).

Fourth reason for acquiring an IDPS is threat documentation. The implementation of security technology usually requires that project proponents document the threat from which the organization must be protected. IDPSs are one means of collecting such data.
Data collected by an IDPS can also help management with quality assurance and continuous improvement; IDPSs consistently pick up information about attacks that have successfully compromised the outer layers of information security controls such as a firewall. This information can be used to identify and repair emergent or residual flaws in the security and network architectures and thus help the organization expedite its incident response process and make other continuous improvements.
Finally, even if an IDPS fails to prevent an intrusion, it can still assist in the after-attack review by providing information on how the attack occurred, what the intruder accomplished, and which methods the attacker employed. This information can be used to remedy deficiencies and to prepare the organization's network environment for future attacks.

IPS technologies are differentiated from IDS technologies by one characteristic: IPS technologies can respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which can be divided into the following groups:

1. **The IPS stops the attack itself**. Examples of how this could be done are as follows:
   - Terminate the network connection or user session that is being used for the attack
   - Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute
   - Block all access to the targeted host, service, application, or other resource.
2. **The IPS changes the security environment.** The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target and altering a host-based firewall on a target to block incoming attacks.
3. **The IPS changes the attack's content**. Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned e-mail to reach its recipient.

## Types of IDPS

IDPSs operate as network- or host-based systems.

**Network-based IDPS**

A network-based IDPS is focused on protecting network information assets. Two specialized subtypes of network-based IDPS are the wireless IDPS and the network behavior analysis (NBA) IDPS. The wireless IDPS focuses on wireless networks, as the name indicates, while the NBA IDPS examines traffic flow on a network in an attempt to recognize abnormal patterns like DDoS, malware, and policy violations.

Network-based IDPS (NIDPS) resides on a computer or appliance connected to a segment of an organization's network and monitors network traffic on that network segment, looking for indications of ongoing or successful attacks. When the NIDPS identifies activity that it is programmed to recognize as an attack, it responds by sending notifications to administrators.
When examining incoming packets, an NIDPS looks for patterns within network traffic such as large collections of related items of a certain type—which could indicate that a denial-of-service attack is underway—or the exchange of a series of related packets in a certain pattern—which could indicate that a port scan is in progress

A NIDPS is installed at a specific place in the network (such as on the inside of an edge router) from where it is possible to monitor the traffic going into and out of a particular network segment. The NIDPS can be deployed to monitor a specific grouping of host computers on a specific network segment, or it may be installed to monitor all traffic between the systems that make up an entire network. When placed next to a hub, switch, or other key networking device, the NIDPS may use that device's monitoring port. The monitoring port also known as a switched port analysis (SPAN) port or mirror port, is a specially configured connection on a network device that is capable of viewing all of the traffic that moves through the entire device.

Before switches became standard for connecting networks in a shared-collision domain, hubs were used. Hubs receive traffic from one node and retransmit it to all other nodes. This configuration allows any device connected to the hub to monitor all traffic passing through the hub. Unfortunately, it also represents a security risk, since anyone connected to the hub can monitor all the traffic that moves through that network segment. Switches, on the other hand, create dedicated point-to-point links between their ports. These links create a higher level of transmission security and privacy and effectively prevent anyone from capturing, and thus eavesdropping on, the traffic passing through the switch. Unfortunately, the ability to capture the traffic is necessary for the use of an IDPS. Thus, monitoring ports are required.

To determine whether an attack has occurred or is underway, NIDPSs compare measured activity to known signatures in their knowledge base. This is accomplished by means of a special implementation of the TCP/IP stack that reassembles the packets and applies protocol stack verification, application protocol verification, or other verification and comparison techniques.

In the **process of protocol stack verification**, the NIDPSs look for invalid data packets—that is, packets that are malformed under the rules of the TCP/IP protocol. A data packet is verified when its configuration matches one that is defined by the various Internet protocols. The elements of these protocols (IP, TCP, UDP, and application layers such as HTTP) are combined in a complete set called the protocol stack when the software is implemented in an operating system or application. Many types of intrusions, especially DoS and DDoS attacks, rely on the creation of improperly formed packets to take advantage of weaknesses in the protocol stack in certain operating systems or applications.

In **application protocol verification**, the higher-order protocols (HTTP, FTP, and Telnet) are examined for unexpected packet behavior or improper use. Sometimes an attack uses valid protocol packets but in excessive quantities (in the case of the tiny fragment attack, the packets are also excessively fragmented). While the protocol stack verification looks for violations in the protocol packet structure, the application protocol verification looks for violations in the protocol packet's use.

**The advantages of NIDPSs include the following:**

1. Good network design and placement of NIDPS devices can enable an organization to use a few devices to monitor a large network.
2. NIDPSs are usually passive devices and can be deployed into existing networks with little or no disruption to normal network operations.
3. NIDPSs are not usually susceptible to direct attack and, in fact, may not be detectable by attackers

**The disadvantages of NIDPSs include the following:**

1. A NIDPS can become overwhelmed by network volume and fail to recognize attacks it might otherwise have detected.
2. NIDPSs require access to all traffic to be monitored. The broad use of switched Ethernet networks has replaced the ubiquity of shared collision domain hubs. Since many switches have limited or no monitoring port capability, some networks are not capable of providing aggregate data for analysis by a NIDPS.
3. NIDPSs cannot analyze encrypted packets, making some of the network traffic invisible to the process. The increasing use of encryption that hides the contents of some or all of the packet by some network services (such as SSL, SSH, and VPN) limits the effectiveness of NIDPSs.
4. NIDPSs cannot reliably ascertain if an attack was successful or not. This requires the network administrator to be engaged in an ongoing effort to evaluate the results of the logs of suspicious network activity.
5. Some forms of attack are not easily discerned by NIDPSs, specifically those involving fragmented packets.

- **Wireless NIDPS**. A wireless IDPS monitors and analyzes wireless network traffic, looking for potential problems with the wireless protocols (Layers 2 and 3 of the OSI model). Unfortunately, wireless IDPSs cannot evaluate and diagnose issues with higher-layer protocols like TCP and UDP. Wireless IDPS capability can be built into a device that provides a wireless access point.

Sensor locations for wireless networks can be located at the access points, on specialized sensor components, or incorporated into selected mobile stations. Centralized management stations collect information from these sensors.

**Some issues associated with the implementation of wireless IDPSs include:**

- **Physical security:** Unlike wired network sensors, which can be physically secured, many wireless sensors are located in public areas like conference rooms, assembly areas, and hallways in order to obtain the widest possible network range. Some of these locations may even be outdoors, as more and more organization are deploying networks in external locations. Thus the physical security of these devices is an issue, which may likely require additional security configuration and monitoring.
- **Sensor range:** A wireless device's range can be affected by atmospheric conditions, building construction, and the quality of both the wireless network card and access point. Some IDPS tools allow an organization to identify the optimal location for sensors by modeling the wireless footprint based on signal strength. Sensors are most effective when their footprints overlap.
- **Access point and wireless switch locations:** Wireless components with bundled IDPS capabilities must be carefully deployed to optimize the IDPS sensor detection grid. The minimum range is just that; you must guard against the possibility of an attacker connecting to a wireless access point from a range far beyond the minimum.
- **Wired network connections:** Wireless network components work independently of the wired network when sending and receiving between stations and access points. However, a network connection eventually integrates wireless traffic with the organization's wired network. Where there is no available wired network connection, it may be impossible to deploy a sensor.
- **Cost**: The more sensors deployed, the more expensive the configuration. Wireless components typically cost more than their wired counterparts, and thus the total cost of ownership of IDPS of both wired and wireless varieties should be carefully considered.

**The wireless IDPS can also detect:**

- Unauthorized WLANs and WLAN devices
- Poorly secured WLAN devices
- Unusual usage patterns
- The use of wireless network scanners Denial of service (DoS) attacks and conditions
- Impersonation and man-in-the-middle attacks

Wireless IDPSs are unable to detect certain passive wireless protocol attacks, in which the attacker monitors network traffic without active scanning and probing. r. By simply looking at wireless devices, which are often visible in public areas, attackers can customdesign evasion methods to exploit the system's channel scanning scheme. Wireless IDPSs can protect the WLAN with which they are associated, but may be susceptible to logical and physical attacks on the wireless access point or the wireless IDPS devices themselves

- **Network Behavior Analysis System** NBA systems examine network traffic in order to identify problems related to the flow of traffic. They use a version of the anomaly detection method described later in this section to identify excessive packet flows such as might occur in the case of equipment malfunction, DoS attacks, virus and worm attacks, and some forms of network policy violations. NBA IDPSs typically monitor internal networks but occasionally monitor connections between internal and external networks.
  **Typical flow data particularly relevant to intrusion detection and prevention includes:**
  - Source and destination IP addresses
  - Source and destination TCP or UDP ports or ICMP types and codes
  - Number of packets and bytes transmitted in the session
  - Starting and ending timestamps for the session

Most NBA sensors can be deployed in passive mode only, using the same connection methods (e.g., network tap, switch spanning port) as network-based IDPS.

**Passive sensors** that are performing direct network monitoring should be placed so that they can monitor key network locations, such as the divisions between networks, and key network segments, such as demilitarized zone (DMZ) subnets.

**Inline sensors** are typically intended for network perimeter use, so they would be deployed in close proximity to the perimeter firewalls, often between the firewall and the Internet border router to limit incoming attacks that could overwhelm the firewall.

**The types of events most commonly detected by NBA sensors include the following:**

- DoS attacks (including DDoS attacks)
- Scanning
- Worms
- Unexpected application services (e.g., tunneled protocols, back doors, use of forbidden application protocols)
- Policy violations

**NBA sensors offer various intrusion prevention capabilities, including the following (grouped by sensor type):**

- **Passive only**
- **Ending the current TCP session**. A passive NBA sensor can attempt to end an existing TCP session by sending TCP reset packets to both endpoints.
- **Inline only**
  **Performing inline firewalling**. Most inline NBA sensors offer firewall capabilities that can be used to drop or reject suspicious network activity.
- **Both passive and inline**
  **Reconfiguring other network security devices**. Many NBA sensors can instruct network security devices such as firewalls and routers to reconfigure themselves to block certain types of activity or route it elsewhere, such as a quarantine virtual local area network (VLAN).
  **Running a third-party program or script.** Some NBA sensors can run an administrator-specified script or program when certain malicious activity is detected.

**Host-Based IDPS**

A host-based IDPS protects the server or host's information assets; the example shown in Figure 7-1 monitors both network connection activity and current information states on host servers. The application-based model works on one or more host systems that support a single application and defends that specific application from special forms of attack.

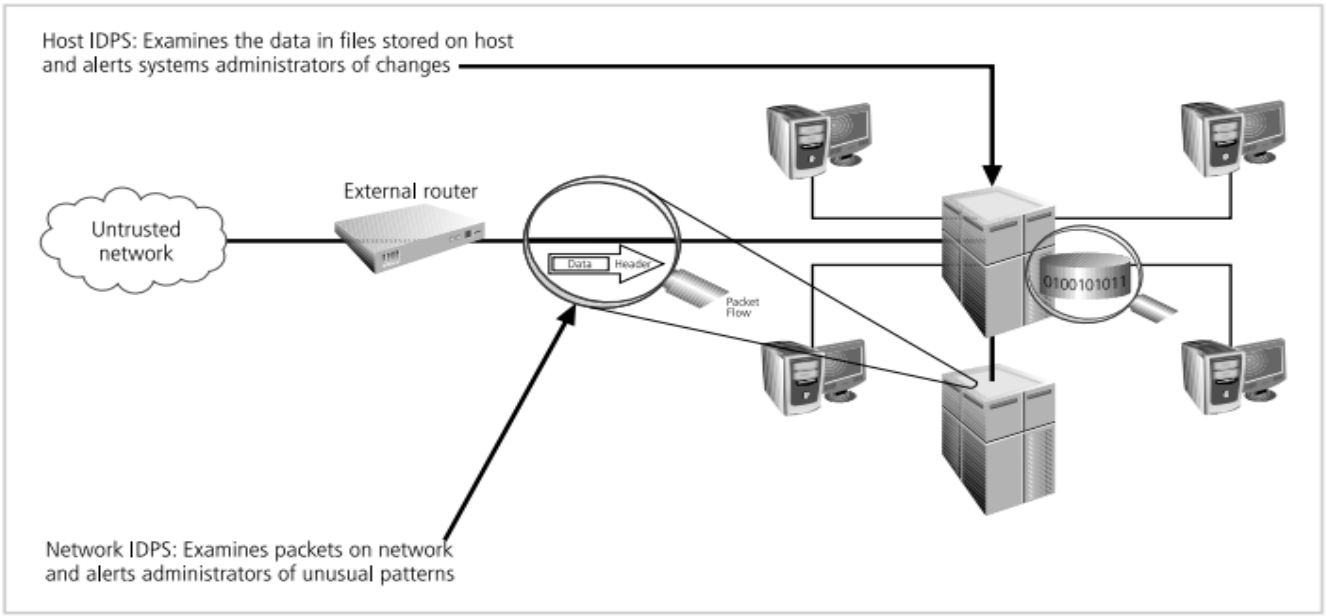


**Figure 7-1** Intrusion Detection and Prevention Systems

While a network-based IDPS resides on a network segment and monitors activities across that segment, a host-based IDPS (HIDPS) resides on a particular computer or server, known as the host, and monitors activity only on that system. HIDPSs are also known as system integrity verifiers11 because they benchmark and monitor the status of key system files and detect when an intruder creates, modifies, or deletes monitored files.

An HIDPS has an advantage over an NIDPS in that it can access encrypted information traveling over the network and use it to make decisions about potential or actual attacks.

An HIDPS is also capable of monitoring system configuration databases, such as Windows registries, in addition to stored configuration files like .ini, .cfg, and .dat files. Most HIDPSs work on the principle of configuration or change management, which means that they record the sizes, locations, and other attributes of system files. The HIDPS triggers an alert when one of the following occurs: file attributes change, new files are created, or existing files are deleted. An HIDPS can also monitor systems logs for predefined events. The HIDPS examines these files and logs to determine if an attack is underway or has occurred and if the attack is succeeding or was successful. The HIDPS maintains its own log file so that an audit trail is available even when hackers modify files on the target system to cover their tracks.

Once properly configured, an HIDPS is very reliable. The only time an HIDPS produces a false positive alert is when an authorized change occurs for a monitored file. This action can be quickly reviewed by an administrator, who may choose to disregard subsequent changes to the same set of files. If properly configured, an HIDPS can also detect when users attempt to modify or exceed their access authorization level.

An HIDPS classifies files into various categories and then sends notifications when changes occur. Most HIDPSs provide only a few general levels of alert notification.

Managed HIDPSs can monitor multiple computers simultaneously by creating a configuration file on each monitored host and by making each HIDPS report back to a master console system, which is usually located on the system administrator's computer. This master console monitors the information provided by the managed hosts and notifies the administrator when it senses recognizable attack conditions.

ne of the most common methods of categorizing folders and files is by color coding. Critical systems components are coded red and usually include the system registry, any folders containing the OS kernel, and application software. Critically important data should also be included in the red category. Support components, such as device drivers and other relatively important files, are generally coded yellow; user data is usually coded green, not because it is unimportant, but because monitoring changes to user data is practically difficult and strategically less urgent

**The advantages of HIDPSs include:**

1. An HIDPS can detect local events on host systems and also detect attacks that may elude a network-based IDPS.
2. An HIDPS functions on the host system, where encrypted traffic will have been decrypted and is available for processing.
3. The use of switched network protocols does not affect an HIDPS.
4. An HIDPS can detect inconsistencies in how applications and systems programs were used by examining the records stored in audit logs. This can enable it to detect some types of attacks, including Trojan horse programs.

**The disadvantages of HIDPSs include:**

1. HIDPSs pose more management issues because they are configured and managed on each monitored host. Operating an HIDPS requires more management effort to install, configure, and operate than does a comparably sized NIDPS solution.
2. An HIDPS is vulnerable both to direct attacks and to attacks against the host operating system. Either circumstance can result in the compromise and/or loss of HIDPS functionality.
3. An HIDPS is not optimized to detect multihost scanning, nor is it able to detect the scanning of non-host network devices, such as routers or switches. Unless complex correlation analysis is provided, the HIDPS will not be aware of attacks that span multiple devices in the network.
4. An HIDPS is susceptible to some denial-of-service attacks.
5. An HIDPS can use large amounts of disk space to retain the host OS audit logs; to function properly, it may be necessary to add disk capacity to the system.
6. An HIDPS can inflict a performance overhead on its host systems, and in some cases may reduce system performance below acceptable levels.

## IDPS Detection Methods

IDPSs use a variety of detection methods to monitor and evaluate network traffic.

**Three methods dominate:**

- **Signature-Based IDPS** A signature-based IDPS (sometimes called a knowledge-based IDPS or a misuse-detection IDPS) examines network traffic in search of patterns that match known signatures—that is, preconfigured, predetermined attack patterns.
  A potential problem with the signature-based approach is that new attack strategies must continually be added into the IDPS's database of signatures; otherwise, attacks that use new strategies will not be recognized and might succeed. Another weakness of the signaturebased method is that a slow, methodical attack might escape detection if the relevant IDPS attack signature has a shorter time frame
- **Statistical Anomaly-Based IDPS** The statistical anomaly-based IDPS (stat IDPS) or behavior-based IDPS collects statistical summaries by observing traffic that is known to be normal. This normal period of evaluation establishes a performance baseline. Once the baseline is established, the stat IDPS periodically samples network activity and, using statistical methods, compares the sampled network activity to this baseline. When the measured activity is outside the baseline parameters—exceeding what is called the clipping level—the IDPS sends an alert to the administrator.
  The advantage of the statistical anomaly-based approach is that the IDPS can detect new types of attacks, since it looks for abnormal activity of any type. Unfortunately, these systems require much more overhead and processing capacity than signature-based IDPSs, because they must constantly compare patterns of activity against the baseline. Another drawback is that these systems may not detect minor changes to system variables and may generate many false positives
- **Stateful Protocol Analysis IDPS** Stateful protocol analysis (SPA) is a process of comparing predetermined profiles of generally accepted definitions of benign activity for each protocol state against observed events to identify deviations. Stateful protocol analysis relies on vendor-developed universal profiles that specify how particular protocols should and should not be used."15 Essentially, the IDPS knows how a protocol, such as FTP, is supposed to work, and therefore can detect anomalous behavior. By storing relevant data detected in a session and then using that data to identify intrusions that involve multiple requests and responses, the IDPS can better detect specialized, multisession attacks. This process is sometimes called deep packet inspection because SPA closely examines packets at the application layer for information that indicates a possible intrusion.
  The models used for SPA are similar to signatures in that they are provided by vendors. These models are based on industry protocol standards established by such entities as the Internet Engineering Task Force, but they vary along with the protocol implementations in such documents.
  nfortunately, the analytical complexity of session-based assessments is the principal drawback to this type of IDPS method, which also requires heavy processing overhead to track multiple simultaneous connections. Additionally, unless a protocol violates its fundamental behavior, this IDPS method may completely fail to detect an intrusion. One final issue is that the IDPS may in fact interfere with the normal operations of the protocol it's examining, especially with client- and server-differentiated operations.
- **Log File Monitors** A log file monitor (LFM) IDPS is similar to a NIDPS. Using LFM, the system reviews the log files generated by servers, network devices, and even other IDPSs, looking for patterns and signatures that may indicate that an attack or intrusion is in process or has already occurred. While an individual host IDPS can only examine the activity in one system, the LFM is able to look at multiple log files from a number of different systems.

## IDPS Response Behavior

Each IDPS responds to external stimulation in a different way, depending on its configuration and function. Some respond in active ways, collecting additional information about the intrusion, modifying the network environment, or even taking action against the intrusion. Others respond in passive ways, for example by setting off alarms or notifications or collecting passive data through SNMP traps.

- **IDPS Response Options** When an IDPS detects a possible intrusion, it has a number of response options, depending on the implementing organization's policy, objectives, and system capabilities. When configuring an IDPS's responses, the system administrator must exercise care to ensure that a response to an attack (or potential attack) does not inadvertently exacerbate the situation.

An analogy to this approach is a car thief who approaches a desirable target in the early a.m., strikes the car with a rolled-up newspaper to trigger the alarm, and then ducks into the bushes. The car owner wakes up, checks the car, determines there is no danger, resets the alarm, and goes back to bed. The thief repeats the triggering action every half hour or so until the owner disables the alarm. The thief is now free to steal the car without worrying about triggering the alarm.
IDPS responses can be classified as active or passive.

- An active response is a definitive action automatically initiated when certain types of alerts are triggered and can include collecting additional information, changing or modifying the environment, and taking action against the intruders.
- Passive response IDPSs simply report the information they have collected and wait for the administrator to act. Generally, the administrator chooses a course of action after analyzing the collected data.

**The following list describes some of the responses an IDPS can be configured to produce**

- **Audible/visual alarm:** The IDPS can trigger a .wav file, beep, whistle, siren, or other audible or visual notification to alert the administrator of an attack. The most common type of such notifications is the computer pop-up, which can be configured with color indicators and specific messages, and can also contain specifics about the suspected attack.
- **SNMP traps and plug-ins:** The Simple Network Management Protocol contains trap functions, which allow a device to send a message to the SNMP management console indicating that a certain threshold has been crossed, either positively or negatively. The IDPS can execute this trap, telling the SNMP console an event has occurred. Some of the advantages of this operation include the relatively standard implementation of SNMP in networking devices, the ability to configure the network system to use SNMP traps in this manner, the ability to use systems specifically to handle SNMP traffic.
- **E-mail message:** The IDPS can send e-mail to notify network administrators of an event. Many administrators use smartphones and other e-mail enabled devices to check for alerts and other notifications frequently. Organizations should use caution in relying on e-mail systems as the primary means of communication between the IDPS and security personnel.
- **Page or phone message:** The IDPS can be configured to dial a phone number and produce an alphanumeric page or a modem noise.
- **Log entry:** The IDPS can enter information about the event (e.g., addresses, time, systems involved, protocol information) into an IDPS system log file or operating system log file. These files can be stored on separate servers to prevent skilled attackers from deleting entries about their intrusions.
- **Evidentiary packet dump:** Organizations that require an audit trail of the IDPS data may choose to record all log data in a special way. This method allows the organization to perform further analysis on the data and also to submit the data as evidence in a civil or criminal case. Once the data has been written using a cryptographic hashing algorithm (discussed in detail in Chapter 8), it becomes evidentiary documentation— that is, suitable for criminal or civil court use. This packet logging can, however, be resource-intensive, especially in denial-of-service attacks.
- **Take action against the intruder:** It has become possible, although not advisable, to take action against an intruder. Known as trap-and-trace, back-hacking, or traceback, this response option involves configuring intrusion detection systems to trace the data from the target system to the attacking system in order to initiate a counterattack. While this may sound tempting, it is ill-advised and may not be legal. An organization only owns a network to its perimeter, and conducting traces or back-hacking to sys□tems outside that perimeter may make the organization just as criminally liable as the individual(s) who began the attack.
- **Launch program:** An IDPS can be configured to execute a specific program when it detects specific types of attacks. A number of vendors have specialized tracking, tracing, and response software that can be part of an organization's intrusion response strategy.
- **Reconfigure firewall:** An IDPS can send a command to the firewall to filter out sus□pected packets by IP address, port, or protocol. While it may not be easy, an IDPS can block or deter intrusions via one of the following methods:
  - Establishing a block for all traffic from the suspected attacker's IP address, or even from the entire source network from which the attacker appears to be operating. This blocking can be set for a specific period of time and reset to normal rules after that period has expired.
  - Establishing a block for specific TCP or UDP port traffic from the suspected attacker's address or source network, blocking only the services that seem to be under attack.

- Blocking all traffic to or from a network interface (such as the organization's Inter□net connection) if the severity of the suspected attack warrants that level of response.
- Terminate session: Terminating the session by using the TCP/IP protocol specified packet TCP close is a simple process. Some attacks would be deterred or blocked by session termination, but others would simply continue when the attacker issues a new session request.
- Terminate connection: The last resort for an IDPS under attack is to terminate the organization's internal or external connections. Smart switches can cut traffic to or from a specific port, should that connection be linked to a system that is malfunctioning or otherwise interfering with efficient network operations.

- **Reporting and Archiving Capabilities** Many, if not all, commercial IDPSs can generate routine reports and other detailed information documents, such as reports of system events and intrusions detected over a particular reporting period (for example, a week or a month). Some provide statistics or logs in formats suitable for inclusion in database systems or for use in report generating packages.
- **Failsafe Considerations for IDPS Responses** Failsafe features protect an IDPS from being circumvented or defeated by an attacker. There are several functions that require failsafe measures. For instance, IDPSs need to provide silent, reliable monitoring of attackers. Should the response function of an IDPS break this silence by broadcasting alarms and alerts in plaintext over the monitored network, attackers can detect the IDPS and might then directly target it in the attack. Encrypted tunnels or other cryptographic measures that hide and authenticate communications are excellent ways to secure and ensure the reliability of the IDPS.

## Selecting IDPS Approaches and Products

The wide array of available intrusion detection products addresses a broad range of organizational security goals and considerations; the process of selecting products that represent the best fit for any particular organization is challenging. The following considerations and questions may help you prepare a specification for acquiring and deploying an intrusion detection product.

- **Technical and Policy Considerations** In order to determine which IDPS best meets an organization's needs, first consider the organizational environment in technical, physical, and political terms.
  - **What Is Your Systems Environment?** The first requirement for a potential IDPS is that it function in your systems environment. This is important; if an IDPS is not designed to accommodate the information sources that are available on your systems, it will not be able to see anything—neither normal activity nor an attack—on your systems
    - What are the technical specifications of your systems environment?
    - What are the technical specifications of your current security protections?
    - What are the goals of your enterprise?
    - How formal is the system environment and management culture in your organization?

  - **What Are Your Security Goals and Objectives?** The next step is to articulate the goals and objectives you wish to attain by using an IDPS.
    - Is the primary concern of your organization protecting from threats originating outside your organization?
    - Is your organization concerned about insider attack?
    - Does your organization want to use the output of your IDPS to determine new needs?
    - Does your organization want to use an IDPS to maintain managerial control (non□security related) over network usage?

  - **What Is Your Existing Security Policy?** You should review your existing organization security policy, which will serve as the template against which your IDPS will be configured. You may find you need to augment the policy, or else derive the following items from it.
    - How is it structured?
    - What are the general job descriptions of your system users?
    - Does the policy include reasonable use policies or other management provisions?
    - Has your organization defined processes for dealing with specific policy violations?

- **Organizational Requirements and Constraints** Your organization's operational goals, constraints, and culture will affect the selection of the IDPS and other security tools and technologies to protect your systems. Consider the following organizational requirements and limitations.
  - What Requirements Are Levied from Outside the Organization?
    - Is your organization subject to oversight or review by another organization?
    - Are there requirements for public access to information on your organization's systems?
    - Are there other security-specific requirements levied by law? Are there legal requirements for protection of personal information (such as earnings information or medical records) stored on your systems?
    - Are there legal requirements for investigation of security violations that divulge or endanger that information?
    - Are there internal audit requirements for security best practices or due diligence? Do any of these audit requirements specify functions that the IDPSs must provide or support?
    - Is the system subject to accreditation? If so, what is the accreditation authority's requirement for IDPSs or other security protection?
    - Are there requirements for law enforcement investigation and resolution of security incidents? Do they require any IDPS functions, especially having to do with collection and protection of IDPS logs as evidence?

  - What Are Your Organization's Resource Constraints? IDPSs can protect the systems of an organization, but at a price. It makes little sense to incur additional expense for IDPS features if your organization does not have sufficient systems or personnel to handle the alerts they will generate.
    - What is the budget for acquisition and life cycle support of intrusion detection hard□ware, software, and infrastructure?
    - Is there sufficient existing staff to monitor an intrusion detection system full time?
    - Does your organization have authority to instigate changes based on the findings of an intrusion detection system?

- **IDPSs Product Features and Quality** It's important to carefully evaluate any IDPS product by considering the following questions:
  - **Is the Product Sufficiently Scalable for Your Environment?** Many IDPSs cannot function within large or widely distributed enterprise network environments.
  - **How Has the Product Been Tested?** Simply asserting that an IDPS has certain capabilities is not sufficient demonstration that those capabilities are real. You should request demonstrations of a particular IDPS to evaluate its suitability for your environment and goals.
    - Has the product been tested against functional requirements?
    - Has the product been tested against attack?

  - **What Is the User Level of Expertise Targeted by the Product?** Different IDPS vendors target users with different levels of technical and security expertise. Ask the vendor what their assumptions are regarding the users of their products.
  - **Is the Product Designed to Evolve as the Organization Grows?** One important product design goal is the ability to adapt to your needs over time.
    - Can the product adapt to growth in user expertise?
    - Can the product adapt to growth and change of the organization's systems infrastructure?
    - Can the product adapt to growth and change in the security threat environment?

  - **What Are the Support Provisions for the Product?** Like other systems, IDPSs require maintenance and support over time. These needs should be identified in a written report.
    - What are the commitments for product installation and configuration support?
    - What are the commitments for ongoing product support?
    - How often are subscriptions updated?
    - Are there any guarantees associated with the IDPS?
    - What training resources does the vendor provide?
    - What additional training resources are available from the vendor and at what cost?

## Strengths and Limitations of IDPSs

- **Strengths of Intrusion Detection and Prevention Systems** Intrusion detection and prevention systems perform the following functions well:
  - Monitoring and analysis of system events and user behaviours
  - Testing the security states of system configurations
  - Baselining the security state of a system, then tracking any changes to that baseline
  - Recognizing patterns of system events that correspond to known attacks Recognizing patterns of activity that statistically vary from normal activity
  - Managing operating system audit and logging mechanisms and the data they generate
  - Alerting appropriate staff by appropriate means when attacks are detected
  - Measuring enforcement of security policies encoded in the analysis engine
  - Providing default information security policies
  - Allowing non-security experts to perform important security monitoring functions
- **Limitations of Intrusion Detection and Prevention Systems** Intrusion detection systems cannot perform the following functions:
  - Compensating for weak or missing security mechanisms in the protection infrastructure, such as firewalls, identification and authentication systems, link encryption systems, access control mechanisms, and virus detection and eradication software
  - Instantaneously detecting, reporting, and responding to an attack when there is a heavy network or processing load
  - Detecting newly published attacks or variants of existing attacks
  - Effectively responding to attacks launched by sophisticated attackers
  - Automatically investigating attacks without human intervention
  - Resisting all attacks that are intended to defeat or circumvent them
  - Compensating for problems with the fidelity of information sources
  - Dealing effectively with switched networks

Some forms of attacks, conducted by attackers called IDPS terrorists, are designed to trip the organization's IDPS, essentially causing the organization to conduct its own DoS attack by overreacting to an actual, but insignificant, attack.

## Deployment and Implementation of an IDPS

Deploying and implementing an IDPS is not always a straightforward task. The strategy for deploying an IDPS should take into account a number of factors, the foremost being how the IDPS will be managed and where it should be placed. These factors determine the number of administrators needed to install, configure, and monitor the IDPS, as well as the number of management workstations, the size of the storage needed for retention of the data generated by the systems, and the ability of the organization to detect and respond to remote threats.

- **IDPS Control Strategies** An IDPS can be implemented via one of three basic control strategies. A control strategy determines how an organization supervises and maintains the configuration of an IDPS. It also determines how the input and output of the IDPS is managed. The three commonly utilized control strategies are centralized, partially distributed, and fully distributed.
  - **Centralized Control Strategy** As illustrated in Figure 7-4, in a centralized IDPS control strategy all IDPS control functions are implemented and managed in a central location, represented in the figure with the large square symbol labeled "IDPS Console." The IDPS console includes the management software, which collects information from the remote sensors (tri□angular symbols in the figure), analyzes the systems or networks, and determines whether the current situation has deviated from the preconfigured baseline. All reporting features are implemented and managed from this central location. The primary advantages of this strategy are cost and control. With one central implementation, there is one management system, one place to go to monitor the status of the systems or networks, one location for reports, and one set of administrative management.

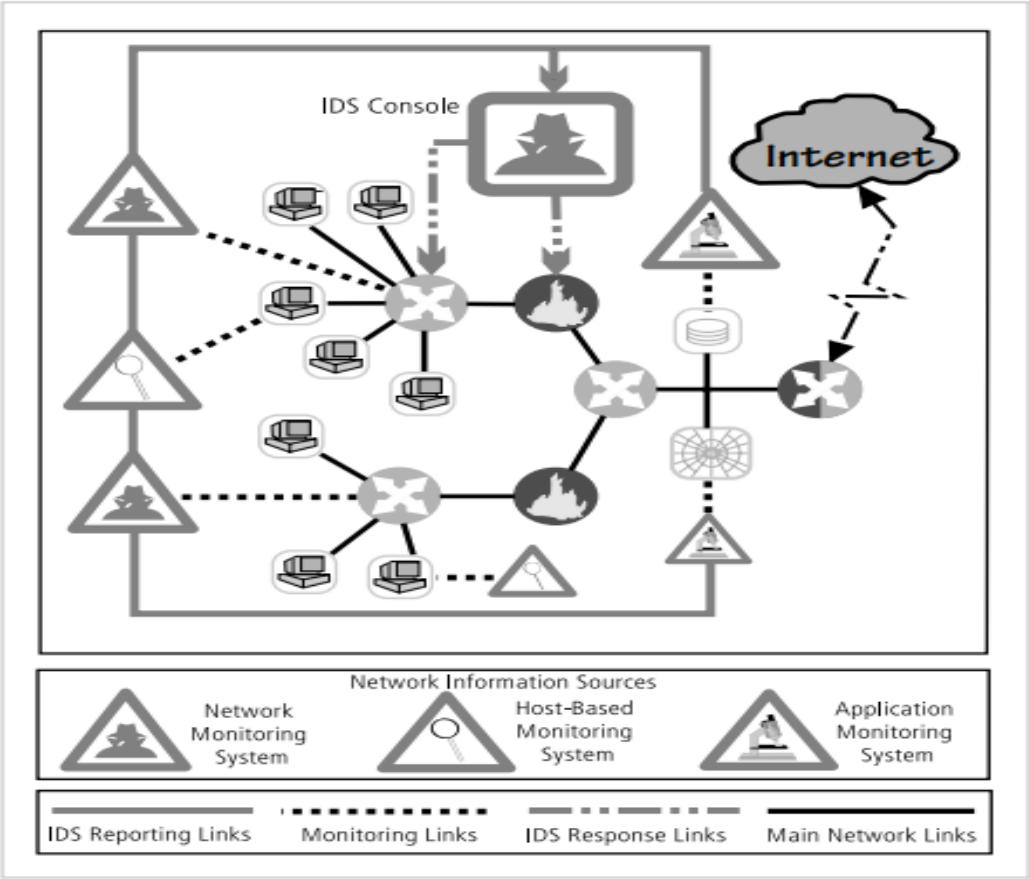


**Figure 7-4** Centralized IDPS Control[13]

- **Fully Distributed Control Strategy** A fully distributed IDPS control strategy, illustrated in Figure 7-5, is the opposite of the centralized strategy. All control functions (which appear in the figure as small square symbols enclosing a computer icon) are applied at the physical location of each IDPS component. Each monitoring site uses its own paired sensors to perform its own control functions to achieve the necessary detection, reaction, and response functions. Thus, each sensor/agent is best configured to deal with its own environment. Since the IDPSs do not have to wait for a response from a centralized control facility, their response time to individual attacks is greatly enhanced.

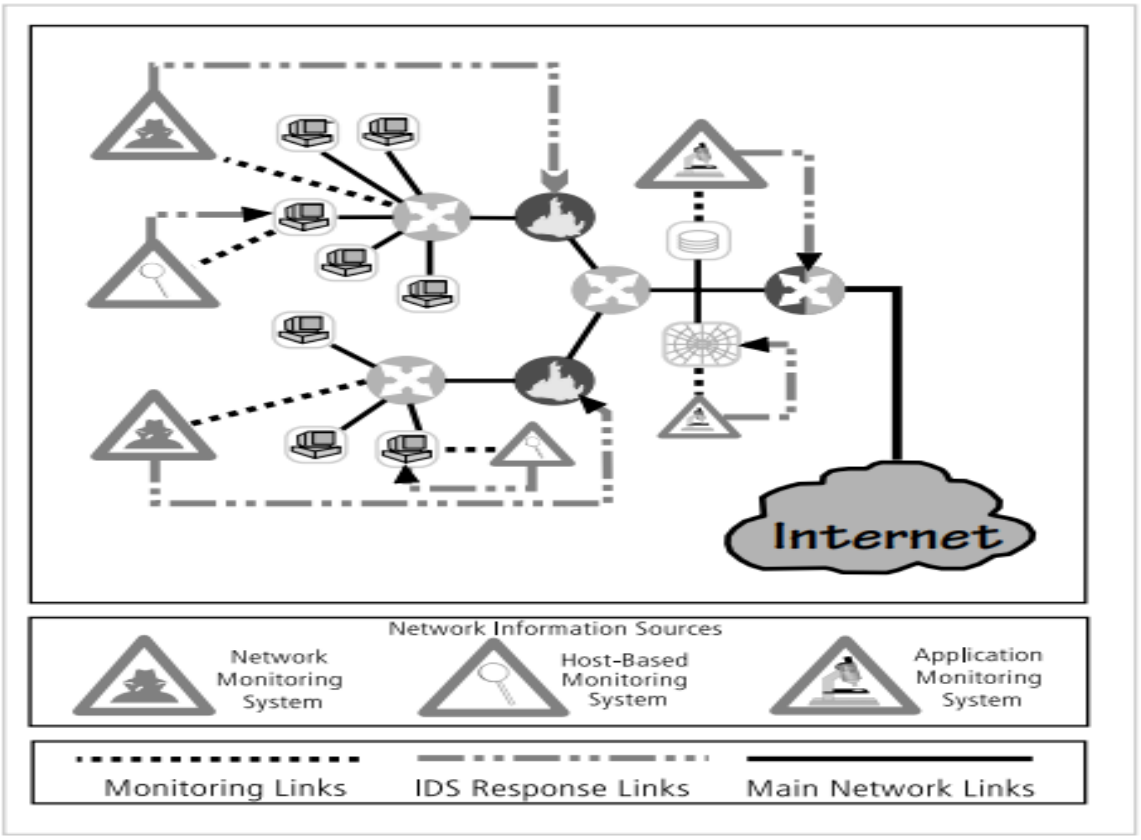


**Figure 7-5** Fully Distributed IDPS Control[14]

➢ **Partially Distributed Control Strategy** A partially distributed IDPS control strategy, depicted in Figure 7-6, combines the best of the other two strategies. While the individual agents can still analyze and respond to local threats, their reporting to a hierarchical central facility enables the organization to detect widespread attacks. This blended approach to reporting is one of the more effective methods of detecting intelligent attackers, especially those who probe an organization at multiple points of entry, trying to identify the systems' configurations and weaknesses, before they launch a concerted attack. The partially distributed control strategy also allows the organization to optimize for economy of scale in the implementation of key management software and personnel, especially in the reporting areas. When the organization can create a pool of security managers to evaluate reports from multiple distributed IDPS systems, it becomes better able to detect these distributed attacks before they become unmanageable.

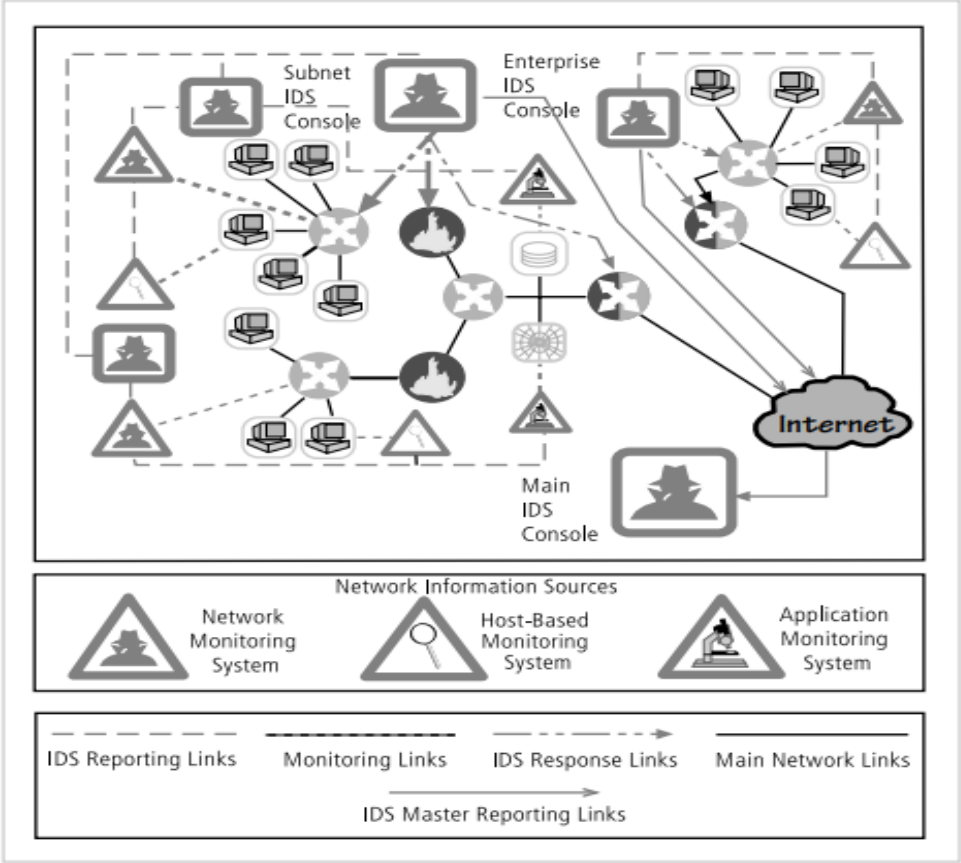


**Figure 7-6** Partially Distributed IDPS Control[15]

- **DPS Deployment** Given the highly technical skills required to implement and configure IDPSs and the imperfection of the technology, great care must be taken when deciding where to locate the components, both in their physical connection to the network and host devices and in how they are logically connected to each other and the IDPS administration team. Since IDPSs are designed to detect, report, and even react to anomalous stimuli, placing IDPSs in an area where such traffic is common can result in excessive reporting.

➢ **Deploying Network-Based IDPSs** The placement of the sensor agents is critical to the operation of all IDPSs, and is especially critical in the case of NIDPSs. NIST recommends the following four locations for NIDPS sensors:

- Location 1: Behind each external firewall, in the network DMZ
  **Advantages:**
  - ❖ IDPS sees attacks that originate from the outside that may penetrate the network's perimeter defenses.
  - ❖ DPS can identify problems with the network firewall policy or performance.
  - ❖ IDPS sees attacks that might target the Web server or FTP server, both of which commonly reside in this DMZ.
- Location 2: Outside an external firewall (See Figure 7-7, location 2)
  **Advantages**:
  - ❖ IDPS documents the number of attacks originating on the Internet that target the network.

- ❖ IDPS documents the types of attacks originating on the Internet that target the network.

- Location 3: On major network backbones (See Figure 7-7, location 3)
  **Advantages**:
  - ❖ IDPS monitors a large amount of a network's traffic, thus increasing its chances of spotting attacks.
  - ❖ IDPS detects unauthorized activity by authorized users within the organization's secu□rity perimeter
- Location 4: On critical subnets (See Figure 7-7, location 4)
  **Advantages:**
  - ❖ IDPS detects attacks targeting critical systems and resources.
  - ❖ This location allows organizations with limited resources to focus these resources on the most valuable network assets.

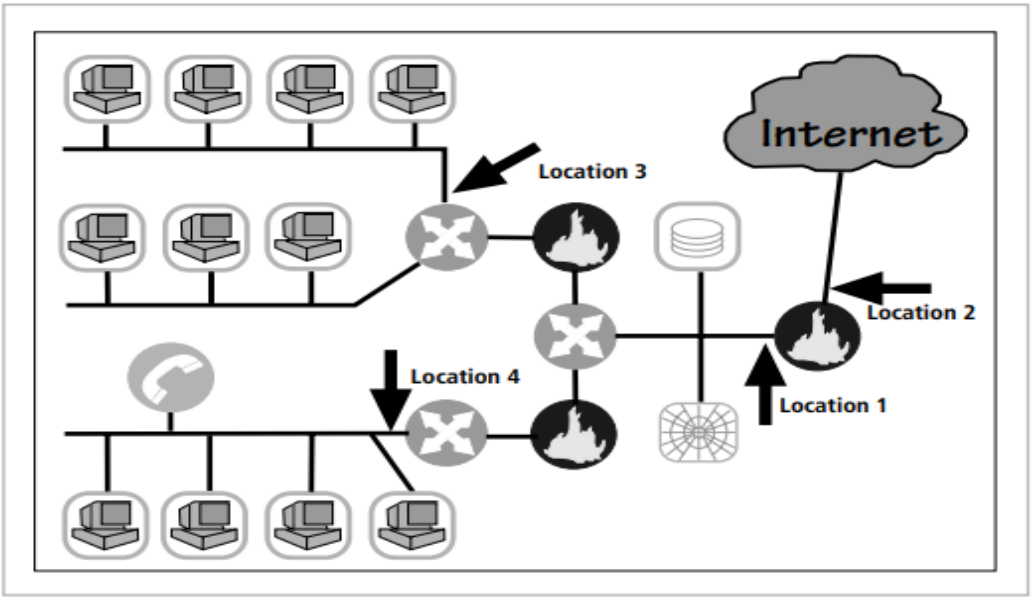


**Figure 7-7** Network IDPS Sensor Locations[17]

- **Deploying Host-Based IDPSs** The proper implementation of HIDPSs can be a painstaking and time-consuming task, as each HIDPS must be custom configured to its host systems. Deployment begins with implementing the most critical systems first. This poses a dilemma for the deployment team, since the first systems to be implemented are mission-critical, and any problems in the installation could be catastrophic to the organization. Thus it may be beneficial to practice an implementation on one or more test servers configured on a network segment that resembles the mission-critical systems. Installation continues until all systems are installed or the organization reaches the planned degree of coverage it is willing to live with, in terms of the number of systems or percent□age of network traffic. To provide ease of management, control, and reporting, each HIDPS should, as discussed earlier, be configured to interact with a central management console.
  Just as technicians can install the HIDPS in offline systems to develop expertise and identify potential problems, users and managers can gain expertise and understanding of the operation of the HIDPS by using a test facility. This test facility could use the offline systems con□figured by the technicians but also be connected to the organization's backbone to allow the HIDPS to process actual network traffic. This setup will also enable technicians to create a baseline of normal traffic for the organization.

## Measuring the Effectiveness of IDPSs

When selecting an IDPS one typically looks at the following four measures of comparative effectiveness:

- **Thresholds:** A threshold is a value that sets the limit between normal and abnormal behavior. Thresholds usually specify a maximum acceptable level, such as x failed connection attempts in 60 seconds, or x characters for a filename length. Thresholds are most often used for anomaly-based detection and stateful protocol analysis.
- **Blacklists and whitelists**: A blacklist is a list of discrete entities, such as hosts, TCP or UDP port numbers, ICMP types and codes, applications, usernames, URLs, filenames, or file extensions, that have been associated with malicious activity. Whitelists and blacklists are most commonly used in signature-based detection and stateful protocol analysis.

Blacklists, also known as hot lists, typically allow IDPSs to block activity that is highly likely to be malicious, and may also be used to assign a higher priority to alerts that match blacklist entries.
A whitelist is a list of discrete entities that are known to be benign. Whitelists are typically used on a granular basis, such as protocol-by-protocol, to reduce or ignore false positives involving known benign activity from trusted hosts.

- **Alert settings**: Most IDPS technologies allow administrators to customize each alert type. Examples of actions that can be performed on an alert type include:
  - Toggling it on or off
  - Setting a default priority or severity level
  - Specifying what information should be recorded and what notification methods (e.g., e-mail, pager) should be used
  - Specifying which prevention capabilities should be used S

  Some products also suppress alerts if an attacker generates many alerts in a short period of time and may also temporarily ignore all future traffic from the attacker. This is to prevent the IDPS from being overwhelmed by alerts.

- Code viewing and editing: Some IDPS technologies permit administrators to see some or all of the detection-related code. This is usually limited to signatures, but some technologies allow administrators to see additional code, such as programs used to perform stateful protocol analysis.

Once implemented, IDPSs are evaluated using two dominant metrics: first, administrators evaluate the number of attacks detected in a known collection of probes; second, the administrators examine the level of use, commonly measured in megabits per second of network traffic, at which the IDPSs fail. An evaluation of an IDPS might read something like this: at 100 Mb/s, the IDPS was able to detect 97 percent of directed attacks. This is a dramatic change from the previous method used for assessing IDPS effectiveness, which was based on the total number of signatures the system was currently running—a sort of "more is better" approach. IDPSs use simple signature-based detection. Some systems, use the almost infinite combination of network performance characteristics of statistical-anomaly-based detection to detect a potential attack. Also, some more sophisticated signature-based systems actually use fewer signatures or rules than older, simpler versions—which, in direct contrast to the signature-based assessment method, suggests that less may actually be more. The recognition that the size of the signature base is an insufficient measure of an IDPS's effectiveness led to the development of stress test measurements for evaluating IDPS performance. These only work, however, if the administrator has a collection of known negative and positive actions that can be proven to elicit a desired response. Since developing this collection can be tedious, most IDPS vendors provide testing mechanisms that verify that their systems are performing as expected. Some of these testing processes will enable the administrator to do the following:

- Record and retransmit packets from a real virus or worm scan
- Record and retransmit packets from a real virus or worm scan with incomplete TCP/IP session connections (missing SYN packets)
- Conduct a real virus or worm attack against a hardened or sacrificial system

One note of caution: There is a strong tendency among IDPS administrators to use common vulnerability assessment tools, like Nmap or Nessus, to evaluate the capabilities of an IDPS. While this may seem like a good idea, it will not work as expected, because most IDPS systems are equipped to recognize the differences between a locally implemented vulnerability assessment tool and a true attack.

In order to perform a true assessment of the effectiveness of IDPS systems, the test process should be as realistic as possible in its simulation of an actual event. This means coupling realistic traffic loads with realistic levels of attacks. You cannot expect an IDPS to respond to a few packet probes as if they represent a denial-of-service attack.

## Honeypots, Honeynets, and Padded Cell Systems

A class of powerful security tools that go beyond routine intrusion detection is known variously as honeypots, honeynets, or padded cell systems.

**Honeypots** are decoy systems designed to lure potential attackers away from critical systems. In the industry, they are also known as decoys, lures, and fly-traps.

When a collection of honeypots connects several honeypot systems on a subnet, it may be called a **honeynet**. A honeypot system (or in the case of a honeynet, an entire subnetwork) contains pseudo-services that emulate well-known services, but is configured in ways that make it look vulnerable to attacks.

In sum, honeypots are designed to do the following:

- Divert an attacker from critical systems
- Collect information about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to document the event and, perhaps, respond

A padded cell is a honeypot that has been protected so that that it cannot be easily compromised—in other words, a hardened honeypot. In addition to attracting attackers with tempting data, a padded cell operates in tandem with a traditional IDPS. When the IDPS detects attackers, it seamlessly transfers them to a special simulated environment where they can cause no harm—the nature of this host environment is what gives the approach the name "padded cell."

**Advantages**

- Attackers can be diverted to targets that they cannot damage.
- Administrators have time to decide how to respond to an attacker
- Attackers' actions can be easily and more extensively monitored, and the records can be used to refine threat models and improve system protections.
- Honeypots may be effective at catching insiders who are snooping around a network.

**Disadvantages**

- The legal implications of using such devices are not well understood.
- Honeypots and padded cells have not yet been shown to be generally useful security technologies.
- An expert attacker, once diverted into a decoy system, may become angry and launch a more aggressive attack against an organization's systems.
- Administrators and security managers need a high level of expertise to use these systems.

**Trap-and-Trace Systems**

Trap-and-trace applications, which are an extension of the attractant technologies. These systems use a combination of techniques to detect an intrusion and then trace it back to its source. The trap usually consists of a honeypot or padded cell and an alarm. . While the intruders are distracted, or trapped, by what they perceive to be successful intrusions, the system notifies the administrator of their presence.

The trace feature is an extension to the honeypot or padded cell approach. The trace—which is similar to caller ID—is a process by which the organization attempts to identify an entity discovered in unauthorized areas of the network or systems. If the intruder is someone inside the organization, the administrators are completely within their power to track the individual and turn him or her over to internal or external authorities. If the intruder is outside the security perimeter of the organization, then numerous legal issues arise.

Trap-and-trace systems seem like an ideal solution. Security is no longer limited to defense. Now security administrators can go on the offense. They can track down the perpetrators and turn them over to the appropriate authorities. Under the guise of justice, some less scrupulous administrators may even be tempted to back hack, or hack into a hacker's system to find out as much as possible about the hacker.

There are more legal drawbacks to trap-and-trace. The trap portion frequently involves the use of honeypots or honeynets. When using honeypots and honeynets, administrators should be careful not to cross the line between enticement and entrapment. Enticement is the act of attracting attention to a system by placing tantalizing information in key locations. Entrapment is the act of luring an individual into committing a crime to get a conviction. Enticement is legal and ethical, whereas entrapment is not.

Administrators should also be wary of the wasp trap syndrome. In this syndrome, a concerned homeowner installs a wasp trap in his back yard to trap the few insects he sees flying about. Because these traps use scented bait, however, they wind up attracting far more wasps than were originally present. Security administrators should keep the wasp trap syndrome in mind before implementing honeypots, honeynets, padded cells, or trap-and-trace systems.

**Active Intrusion Prevention**

Some organizations would like to do more than simply wait for the next attack and implement active countermeasures to stop attacks. One tool that provides active intrusion prevention is known as LaBrea.

- LaBrea is a "sticky" honeypot and IDPS and works by taking up the unused IP address space within a network.
- When LaBrea notes an ARP request, it checks to see if the IP address requested is actually valid on the network. If the address is not currently being used by a real computer or network device, LaBrea pretends to be a computer at that IP address and allows the attacker to complete the TCP/IP connection request, known as the three-way handshake.
- Once the handshake is complete, LaBrea changes the TCP sliding window size to a low number to hold open the TCP connection from the attacker for many hours, days, or even months.
- Holding the connection open but inactive greatly slows down network-based worms and other attacks. It allows the LaBrea system time to notify the system and network administrators about the anomalous behavior on the network

**Scanning and Analysis Tools**

In order to secure a network, it is imperative that someone in the organization knows exactly where the network needs securing. This may sound simple and obvious; however, many companies skip this step. To truly assess the risk within a computing environment, you must deploy technical controls using a strategy of defense in depth, which is likely to include intrusion detection systems (IDSs), active vulnerability scanners, passive vulnerability scanners, automated log analyzers, and protocol analyzers (commonly referred to as sniffers). The IDPS, helps to secure networks by detecting intrusions; the remaining items in the list also help secure networks, but they do this by helping administrators identify where the network needs securing. More specifically, scanner and analysis tools can find vulnerabilities in systems, holes in security components, and unsecured aspects of the network.

Although some information security experts may not perceive them as defensive tools, scanners, sniffers, and other such vulnerability analysis tools can be invaluable because they enable administrators to see what the attacker sees. Some of these tools are extremely complex and others are rather simple. The tools also range from expensive commercial products to free. In the military, there is a long and distinguished history of generals inspecting the troops under their command before battle, walking down the line checking out the equipment and mental preparedness of each soldier. In a similar way, the security administrator can use vulnerability analysis tools to inspect the units (host computers and network devices) under his or her command.

Scanning tools are used as part of an attack protocol to collect information that an attacker would need to launch a successful attack. The **attack protocol** is a series of steps or processes used by an attacker, in a logical sequence, to launch an attack against a target system or network. One of the preparatory parts of the attack protocol is the collection of publicly available information about a potential target, a process known as footprinting. The attacker can attack through web page, Web pages usually contain quantities of information about internal systems, individuals developing Web pages, and other tidbits, which can be used for social engineering attacks. The view source option on most popular Web browsers allows the user to see the source code behind the graphics. A number of details in the source code of the Web page can provide clues to potential attackers and give them insight into the configuration of an internal network, such as the locations and directories for Common Gateway Interface (CGI) script bins and the names or possibly addresses of computers and servers.

For example, consider Company X, which has a large datacenter in Atlanta. The datacenter has been secured, and thus it will be very hard for an attacker to break into it via the Internet. However, the attacker has run a "link:" query on the search engine www.altavista.com and found a small Web server that links to Company X's main Web server. After further investigation, the attacker learns that the small Web server was set up by an administrator at a remote facility and that the remote facility has, via its own leased lines, an unrestricted internal link into Company X's corporate datacenter. The attacker can now attack the weaker site at the remote facility and use this compromised network—which is an internal network—to attack the true target.

To assist in the footprint intelligence collection process, you can use an enhanced Web scanner that, among other things, can scan entire Web sites for valuable pieces of information, such as server names and e-mail addresses. One such scanner is called Sam Spade, the details of which can be found in the program's help file. Sam Spade can also do a host of other scans and probes, such as sending multiple ICMP information requests (pings), attempting to retrieve multiple and cross-zoned DNS queries, and performing network analysis queries. Sam Spade is not, however, considered to be hackerware (or hacker-oriented software), but rather it is a utility that happens to be useful to network administrators and miscreants alike

The next phase of the attack protocol is a data-gathering process called fingerprinting. This is a systematic survey of all of the target organization's Internet addresses (which were collected during the footprinting phase described above); the survey is conducted to identify the network services offered by the hosts in that range.

- **Port Scanners** Port scanning utilities, or port scanners, are tools used by both attackers and defenders to identify (or fingerprint) the computers that are active on a network, as well as the ports and services active on those computers, the functions and roles the machines are fulfilling, and other useful information. These tools can scan for specific types of computers, protocols, or resources, or their scans can be generic. It is helpful to understand the network environment so that you can use the tool most suited to the data collection task at hand. The most popular port scanner is Nmap.
  Why secure open ports? Simply put, an open port can be used by an attacker to send commands to a computer, potentially gain access to a server, and possibly exert control over a networking device. The general rule of thumb is to remove from service or secure any port not absolutely necessary to conducting business. For example, if a business doesn't host Web services, there is no need for port 80 to be available on its servers.

- **Firewall Analysis Tools** Understanding exactly where an organization's firewall is located and what the existing rule sets on the firewall do are very important steps for any security administrator. There are several tools that automate the remote discovery of firewall rules and assist the administrator (or attacker) in analyzing the rules to determine exactly what they allow and what they reject.
  The Nmap tool mentioned earlier has some advanced options that are useful for firewall analysis. The Nmap option called idle scanning (which is run with the -I switch) will allow the Nmap user to bounce your scan across a firewall by using one of the idle DMZ hosts as the initiator of the scan. More specifically, since most operating systems do not use truly random IP packet identification numbers (IP IDs), if there is more than one host in the DMZ and one host uses nonrandom IP IDs, then the attacker can query the server (server X) and obtain the currently used IP ID as well as the known algorithm for incrementing the IP IDs. The attacker can then spoof a packet that is allegedly from server X and destined for an internal IP address behind the firewall. If the port is open on the internal machine, the internal machine replies to server X with a SYN-ACK packet, which forces server X to respond with a TCP RESET packet. In responding with the TCP RESET, server X increments its IP ID number. The attacker can now query server X a second time to see if the IP ID has incremented. If it has, the attacker knows that the internal machine is alive and that the internal machine has the queried service port open. In a nutshell, running the Nmap idle scan allows an attacker to scan an internal network as if he or she were physically located on a trusted machine inside the DMZ.

  Another tool that can be used to analyze firewalls is Firewalk. Written by noted author and network security expert Mike Schiffman, Firewalk uses incrementing Time-To-Live (TTL) packets to determine the path into a network as well as the default firewall policy. Running Firewalk against a target machine reveals where routers and firewalls are filtering traffic to the target host.

  A final firewall analysis tool worth mentioning is HPING, which is a modified ping client. It supports multiple protocols and has a command-line method of specifying nearly any of the ping parameters.
  For instance, you can use HPING with modified TTL values to determine the infrastructure of a DMZ. You can use HPING with specific ICMP flags in order to bypass poorly configured firewalls (i.e., firewalls that allow all ICMP traffic to pass through) and find internal systems.

- **Operating System Detection Tools** Detecting a target computer's operating system is very valuable to an attacker, because once the OS is known, all of the vulnerabilities to which it is susceptible can easily be

determined. There are many tools that use networking protocols to determine a remote computer's OS. One specific tool worth mentioning is XProbe, which uses ICMP to determine the remote OS.
When run, XProbe sends many different ICMP queries to the target host. As reply packets are received, XProbe matches these responses from the target's TCP/IP stack with its own internal database of known responses. Because most OSs have a unique way of responding to ICMP requests, Xprobe is very reliable in finding matches and thus detecting the operating systems of remote computers.

- **Vulnerability Scanners**
  **Active vulnerability scanners** scan networks for highly detailed information. An active scanner is one that initiates traffic on the network in order to determine security holes. As a class, this type of scanner identifies exposed usernames and groups, shows open network shares, and exposes configuration problems and other vulnerabilities in servers.
  An example of a vulnerability scanner is GFI LANguard Network Security Scanner (NSS), which is available as freeware for noncommercial use. Another example of a vulnerability scanner is Nessus, which is a professional freeware utility that uses IP packets to identify the hosts available on the network, the services (ports) they are offering, the operating system and OS version they are running, the type of packet filters and firewalls in use, and dozens of other characteristics of the network.
  Vulnerability scanners should be proficient at finding known, documented holes. But what happens if the Web server is from a new vendor or the application was developed by an internal development team? There is a class of vulnerability scanners called blackbox scanners, or fuzzers. Fuzz testing is a straightforward testing technique that looks for vulnerabilities in a program or protocol by feeding random input to the program or a network running the protocol. Vulnerabilities can be detected by measuring the outcome of the random inputs.
  Nessus scanner has a class of attacks called destructive. If enabled, Nessus attempts common overflow techniques against a target host. Fuzzers or blackbox scanners and Nessus in destructive mode can be very dangerous tools and should only be used in a lab environment.

  The **passive vulnerability scanner** is one that listens in on the network and determines vulnerable versions of both server and client software. Passive scanners are advantageous in that they do not require vulnerability analysts to get approval prior to testing. These tools simply monitor the network connections to and from a server to obtain a list of vulnerable applications. Furthermore, passive vulnerability scanners have the ability to find client-side vulnerabilities that are typically not found by active scanners.

- **Packet Sniffers** Another tool worth mentioning is the packet sniffer. A packet sniffer (sometimes called a network protocol analyzer) is a network tool that collects copies of packets from the network and analyzes them. It can provide a network administrator with valuable information for diagnosing and resolving networking issues.
  There are both commercial and open-source sniffers—more specifically, Sniffer is a commercial product, and Snort is open-source software. An excellent free, client-based network protocol analyzer is Wireshark formerly known as Ethereal.
  Wireshark allows the administrator to examine data from both live network traffic and captured traffic. Wireshark has several features, including a language filter and TCP session reconstruction utility. To use these types of programs most effectively, the user must be connected to a network from a central location. Simply tapping into an Internet connection floods you with more data than can be readily processed and technically constitutes a violation of the wiretapping act. To use a packet sniffer legally, the administrator must (1) be on a network that the organization owns, (2) be under direct authorization of the owners of the network, and (3) have knowledge and consent of the content creators. If all three conditions are met, the administrator can selectively collect and analyze packets to identify and diagnose problems on the network. Conditions one and two are self-explanatory. The third, consent, is usually handled by having all system users sign a release when they are issued a user ID and passwords.

  Many administrators feel that they are safe from sniffer attacks when their computing environment is primarily a switched network environment. This couldn't be farther from the truth. There are a number of open-source sniffers that support alternate networking approaches that can, in turn, enable packet sniffing in a switched network environment. Two of these alternate networking approaches are ARP-spoofing and session hijacking (which uses tools like ettercap).

- **Wireless Security Tools** A wireless connection, while convenient, has many potential security holes. An organization that spends all of its time securing the wired network and leaves wireless networks to operate in any manner is opening itself up for a security breach. As a security professional, you must assess the risk of wireless networks. A wireless security toolkit should include the ability to sniff wireless traffic, scan wireless hosts, and assess the level of privacy or confidentiality afforded on the wireless network.
  **Top five wireless tools.**
  - Kismet, a powerful wireless sniffer, network detector, and IDPS, which works by passively sniffing the networks
  - Netstumbler, a freeware Windows destumbler
  - Aircrack, a WEP/WPA cracking tool
  - Airsnort, an 802.11 WEP encryption cracking tool
  - KisMac, a GUI passive wireless stumbler for Mac OS X (variation of Kismet

NetStumbler is offered as freeware. AirSnare is a free tool that can be run on a low-end wireless workstation. AirSnare monitors the airwaves for any new devices or access points. When it finds one, AirSnare sounds an alarm alerting the administrators that a new, potentially dangerous, wireless apparatus is attempting access on a closed wireless network.

## Biometric access control

Biometric access control is based on the use of some measurable human characteristic or trait to authenticate the identity of a proposed systems user (a supplicant). It relies upon recognition—the same thing you rely upon to identify friends, family, and other people you know. The use of biometric-based authentication is expected to have a significant impact in the future as technical and ethical issues with the technology are resolved.

**Biometric authentication technologies include the following:**

- Fingerprint comparison of the supplicant's actual fingerprint to a stored fingerprint
- Palm print comparison of the supplicant's actual palm print to a stored palm print
- Hand geometry comparison of the supplicant's actual hand to a stored measurement
- Facial recognition using a photographic ID card, in which a human security guard compares the supplicant's face to a photo
- Facial recognition using a digital camera, in which a supplicant's face is compared to a stored image
- Retinal print comparison of the supplicant's actual retina to a stored image
- Iris pattern comparison of the supplicant's actual iris to a stored image

Among all possible biometrics, only three human characteristics are usually considered truly unique. They are as follows:

- Fingerprints
- Retina of the eye (blood vessel pattern)
- Iris of the eye (random pattern of features found in the iris, including freckles, pits, striations, vasculature, coronas, and crypts)

Most of the technologies that scan human characteristics convert these images to some form of minutiae. Minutiae are unique points of reference that are digitized and stored in an encrypted format when the user's system access credentials are created. A problem with this method is that some human characteristics can change over time, due to normal development, injury, or illness, which means that system designers must create fallback or failsafe authentication mechanisms.

Signature and voice recognition technologies are also considered to be biometric access controls measures. Signature recognition has become commonplace. Retail stores use signature recognition, or at least signature capture, for authentication during a purchase. The customer signs a digital pad with a special stylus that captures the signature. The signature is digitized and either saved for future reference, or compared with a signature on a database for validation.

Voice recognition works in a similar fashion in that an initial voiceprint of the user reciting a phrase is captured and stored. Later, when the user attempts to access the system, the authentication process requires the user to speak this same phrase so that the technology can compare the current voiceprint against the stored value.

**Effectiveness of Biometrics**

Biometric technologies are evaluated on three basic criteria: first, the false reject rate, which is the percentage of supplicants who are in fact authorized users but are denied access; second, the false accept rate, which is the percentage of supplicants who are unauthorized users but are granted access; and third, the crossover error rate, which is the level at which the number of false rejections equals the false acceptances

- **False Reject Rate** The false reject rate is the percentage of identification instances in which authorized users are denied access as a result of a failure in the biometric device. This failure is known as a Type I error, this error rate is probably of least concern to security professionals since rejection of an authorized user represents no threat to security.
  Most people have experienced the frustration of having a credit card or ATM card fail to perform because of problems with the magnetic strip. In the field of biometrics, similar problems can occur when a system fails to pick up the various information points it uses to authenticate a prospective user properly.
- **False Accept Rate** The false accept rate is the percentage of identification instances in which unauthorized users are allowed access to systems or areas as a result of a failure in the biometric device. This failure is known as a Type II error, and is unacceptable to security professionals.
- **Crossover Error Rate (CER)** The crossover error rate (CER) is the level at which the number of false rejections equals the false acceptances, and is also known as the equal error rate. This is possibly the most common and important overall measure of the accuracy of a biometric system. Most biometric systems can be adjusted to compensate for both false positive and false negative errors.
  Adjustment to one extreme creates a system that requires perfect matches and results in high false rejects, but almost no false accepts. Adjustment to the other extreme produces low false rejects, but high false accepts.
  The trick is to find the balance between providing the requisite level of security and minimizing the frustration level of authentic users. Thus, the optimal setting is found to be somewhere near the point at which these two error rates are equal; that is, at the crossover error rate or CER.

**Acceptability of Biometrics**

Many biometric systems that are highly reliable and effective are considered somewhat intrusive to users. As a result, many information security professionals, in an effort to avoid confrontation and possible user boycott of the biometric controls, don’t implement them. The order of effectiveness is nearly exactly opposite the order of acceptance.