chapter 7

Security Technology: Intrusion Detection and Prevention Systems, and Other Security Tools

Do not wait; the time will never be just right. *Start where you stand, and work with whatever tools you may have at your command, and better tools will be found as you go along.*

NAPOLEON HILL (1883–1970)
FOUNDER OF *THE SCIENCE OF SUCCESS*

Miller Harrison was going to make them sorry and make them pay. Earlier today, his contract with SLS had been terminated, and he'd been sent home. Oh sure, the big shot manager, Charlie Moody, had said Miller would still get paid for the two weeks remaining in his contract, and that the decision was based on "changes in the project and evolving needs as project work continued," but Miller knew better. He knew he'd been let go because of that know-nothing Kelvin and his simpering lapdog Laverne Nguyen. And now he was going to show them and everyone else at SLS who knew more about security.

Miller knew that the secret to hacking into a network successfully was to apply the same patience, attention to detail, and dogged determination that defending a network required. He also knew that the first step in a typical hacking protocol was footprinting—that is, getting a fully annotated diagram of the network. Miller already had one of these—in a violation of company policy, he had brought a copy home last week when Laverne first started trying to tell him how to do his job.

When they terminated his contract today, Miller's supervisors made him turn in his company laptop and then actually had the nerve to search his briefcase. By then, however, Miller had already stashed all the files and access codes he needed to wage an attack.

To begin, he activated his VPN client to connect to the SLS network from his rented connection at an Internet cafe. He realized almost immediately that Charlie Moody had also confiscated the crypto-token that enabled him to use the VPN for remote access. No problem, Miller thought. If the front door was locked, he would try the back door. He cabled his laptop to the analog phone line, opened up a modem dialing program and typed in the dial-up number for SLS he had gotten from the network administrator last week. After the dialer established the connection, Miller positioned his hands on the keyboard, and then he read the prompt on his monitor:

> *SLS Inc. Company Use Only. Unauthorized use is prohibited and subject to prosecution.*
>
> *Enter Passphrase:*

Apparently the SLS security team had rerouted all dial-up requests to the same RADIUS authentication server that the VPN used. So, he was locked out of the back door too. But Miller moved on to his next option, which was to use another back door of his very own. The back door consisted of a zombie program he'd installed on the company's extranet quality assurance server. No one at SLS worried about securing the QA server since it did not store any production data. In fact, the server wasn't even subject to all the change control procedures that were applied to other systems on the extranet. Miller activated the program he used to remotely control the zombie program and typed in the IP address of the computer running the zombie. No response. He opened up a command window and pinged the zombie. The computer at that address answered each ping promptly, which meant that it was alive and well. Miller checked the zombie's UDP port number and ran an Nmap scan against that single computer for that port. It was closed tight. He cursed the firewall, the policy that controlled it, and the technicians that kept it up to date.

With all of his pre-planned payback cut off at the edge of SLS's network, he decided to continue his hack by going back to the first step—specifically, to perform a detailed fingerprinting of all SLS Internet addresses. Since the front and both back doors were locked, it was time to get a new floor plan. He launched a simple network port scanner on his Linux laptop. He restarted Nmap and configured it to scan the entire IP address range for SLS's extranet. With a single keystroke, he unleashed the port scanner on the SLS network.

LEARNING OBJECTIVES:

Upon completion of this material, you should be able to:

- Identify and describe the categories and operating models of intrusion detection and prevention systems
- Define and describe honeypots, honeynets, and padded cell systems
- List and define the major categories of scanning and analysis tools, and describe the specific tools used within each of these categories
- Explain the various methods of access control, including the use of biometric access mechanisms

Introduction

The protection of an organization's information assets relies at least as much on people as on technical controls, but technical solutions, guided by policy and properly implemented, are an essential component of an information security program. Chapter 6 introduced the subject of security technology and covered some specific technologies, including firewalls, dial-up protection mechanisms, content filtering, and VPNs. This chapter builds on that discussion by describing additional and more advanced technologies—intrusion detection and prevention systems, honeypots, honeynets, padded cell systems, scanning and analysis tools, and access controls—that organizations can use to enhance the security of their information assets.

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. Often, the differences among intrusion types lie with the attacker—some intruders don't care which organizations they harm and prefer to remain anonymous, while others crave notoriety.

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. These actions seek to limit the loss from an intrusion and return operations to a normal state as rapidly as possible. 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). With almost all IDSs, system administrators can choose the configuration of the various alerts and the alarm levels associated with each type of 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." The configurations that enable IDSs to provide customized levels of detection and response are quite complex. 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. The following list of IDPS industry standard terms and definitions is taken from a well-known information security company, TruSecure:

- **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.
- **Noise**: Alarm events that are accurate and noteworthy but that do not pose significant threats to information security. Unsuccessful attacks are the most common source of IDPS noise, and some of these may in fact be triggered by scanning and enumeration tools deployed by network users without intent to do harm.
- **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 (commonly experienced as network probes and other "doorknob rattling" activities)
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[1]

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. Also, because such corrective measures usually require that the administrator install patches and upgrades, they are subject to fluctuations in the administrator's workload. To further complicate the matter, sometimes services known to be

vulnerable cannot be disabled or otherwise protected because they are essential to ongoing operations. At such times—namely, when there is a known vulnerability or deficiency in the system—an IDPS can be set up to detect attacks or attempts to exploit existing weaknesses, and thus it becomes an important part of the strategy of defense in depth.

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). A system capable of detecting the early warning signs of footprinting and fingerprinting functions like a neighborhood watch that spots would-be burglars testing doors and windows, enabling administrators to prepare for a potential attack or to take actions to minimize potential losses from an attack.

A 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. (To collect attack information in support of an IDPS implementation, you can begin with a freeware IDPS tool such as Snort).

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. The IDPS can also provide forensic information that may be useful should the attacker be caught and prosecuted or sued.[2]

> *According to the NIST 800-94 guide,*
>
> *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:*
>
> - *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.*

- *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. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.*
- *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. A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain attacks to be discarded as part of the normalization process.*[3]

Types of IDPS

IDPSs operate as network- or host-based systems. 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.

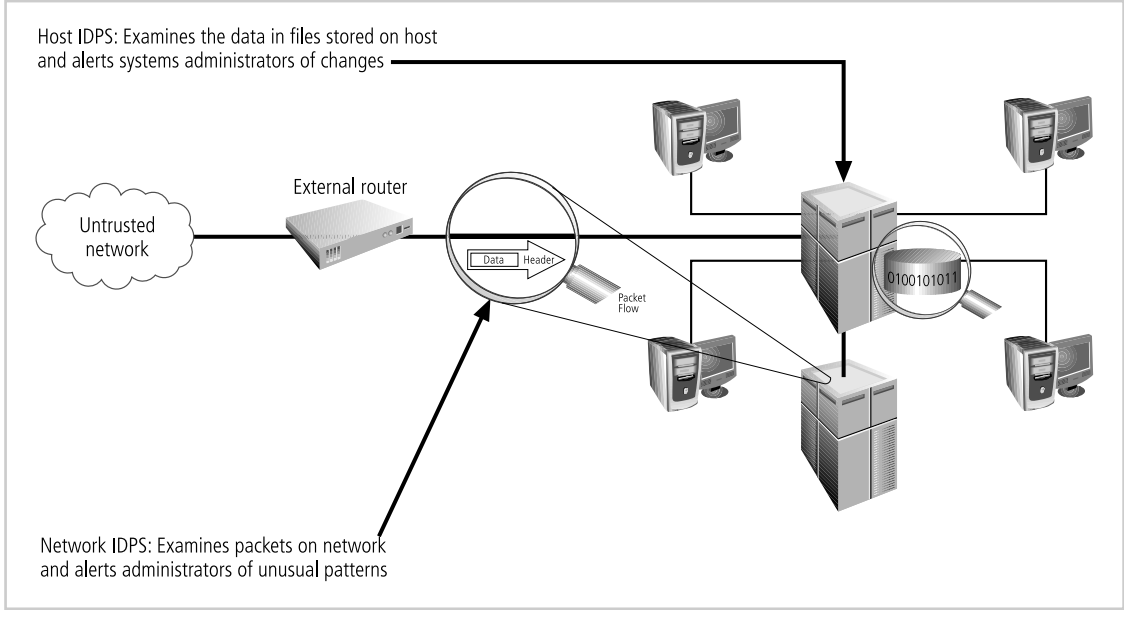

Figure 7-1 Intrusion Detection and Prevention Systems

Source: Course Technology/Cengage Learning

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.

Network-Based IDPS A **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. An NIDPS can detect many more types of attacks than a host-based IDPS, but it requires a much more complex configuration and maintenance program.

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. In the early 1990s, 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. These connections enable network administrators to collect traffic from across the network for analysis by the IDPS as well as for occasional use in diagnosing network faults and measuring network performance.

Figure 7-2 shows data from the Snort Network IDPS Engine (see *www.snort.org*). In this case, the display is a sample screen from Snorby (see *snorby.org*), a client that can manage Snort as well as display the alerts generated.

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

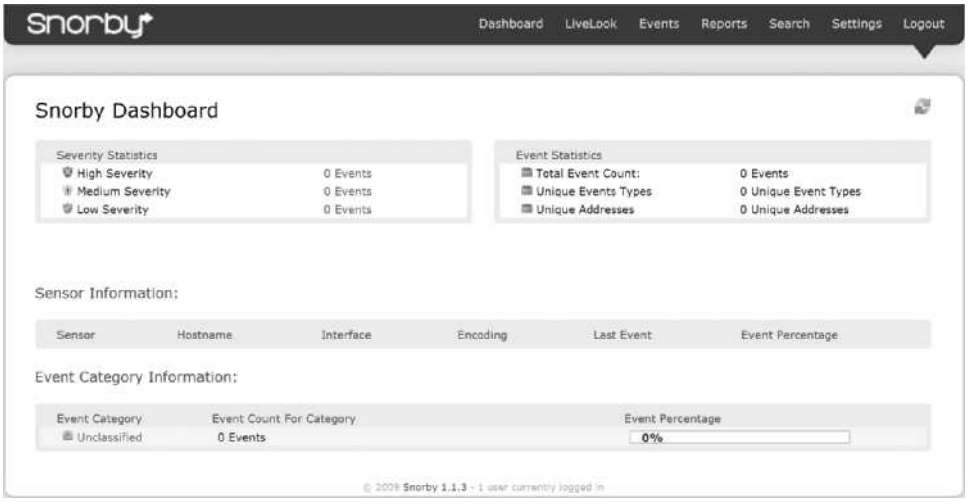

Figure 7-2 Snorby Manages Snort and Displays Alerts
Source: Course Technology/Cengage Learning

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*. One example of this kind of attack is DNS cache poisoning, in which valid packets exploit poorly configured DNS servers to inject false information to corrupt the servers' answers to routine DNS queries from other systems on the network. Unfortunately, this higher-order examination of traffic can have the same effect on an IDPS as it can on a firewall—that is, it slows the throughput of the system. It may be necessary to have more than one NIDPS installed, with one of them performing protocol stack verification and one performing application protocol verification.

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.[4]

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. Some IDPS vendors are accommodating the need for ever faster network performance by improving the processing of detection algorithms in dedicated hardware circuits to gain a performance advantage. Additional efforts to optimize rule set processing may also reduce overall effectiveness in detecting attacks.
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. Even when switches do provide monitoring ports, they may not be able to mirror all activity with a consistent and reliable time sequence.
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. In fact, some NIDPSs are particularly vulnerable to malformed packets and may become unstable and stop functioning.[5]

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, much as other network-based IDPSs do, and aggregate information into a comprehensive assessment of wireless network intrusions. 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.[6]

In addition to the traditional types of intrusions detected by other IDPSs, 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[7]

Wireless IDPSs are unable to detect certain passive wireless protocol attacks, in which the attacker monitors network traffic without active scanning and probing. They are also susceptible to evasion techniques, which are described earlier in this chapter. By simply looking at wireless devices, which are often visible in public areas, attackers can custom-design 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. The best-configured IDPS in the world cannot withstand an attack from a well-placed brick.[8]

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[9]

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 IDPSs.*

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.*[10]

Host-Based IDPS 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 verifiers**[11] 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. Also, since the HIDPS works on only one computer system, all the traffic it examines traverses that system. The packet delivery mode, whether switched or in a shared-collision domain, is not a factor.

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. For example, an administrator can configure an HIDPS to report changes in a system folder (e.g., in C:\Windows or C:\WINNT) and changes to a security-related application (such as C:\TripWire). The configuration rules may classify changes to a specific application folder (e.g., C:\Program Files\Office) as normal and hence unreportable. Administrators can configure the system to log all activity but to page them or e-mail them only if a reportable security event occurs. Since internal application files, such as dictionaries and configuration files, and data files are frequently modified, a poorly configured HIDPS can generate a large volume of false alarms.

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. Figure 7-3 shows a sample screen from Inox Verisys (File Integrity Monitor), a popular HIDPS (see *www.ionx.co.uk*).

One 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. User data files are frequently modified, but systems kernel files, for example, should only be modified during upgrades or installations. If the three-tier system is too simplistic, an organization can use a scale of 0–100, with 100 being most mission-critical and 0 being unimportant. It is not unusual, however, for such systems to result in confusion over issues such as how to respond to level 67 and 68 intrusions. Sometimes simpler is better.

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.

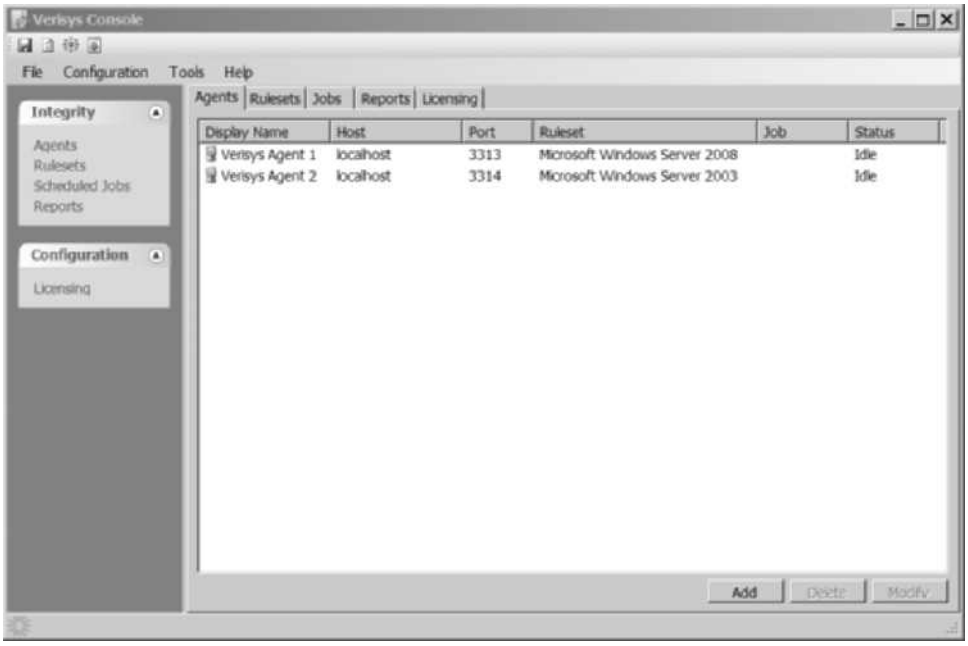

Figure 7-3 Inox Verisys (File Integrity Monitor) HIDPS

Source: Course Technology/Cengage Learning

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.[12]

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.[13]

IDPS Detection Methods

IDPSs use a variety of detection methods to monitor and evaluate network traffic. Three methods dominate: the signature-based approach, the statistical-anomaly approach, and the stateful packet inspection approach.

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. Signature-based IDPS technology is widely used because many attacks have clear and distinct signatures, for example: (1) footprinting and fingerprinting activities use ICMP, DNS querying, and e-mail routing analysis; (2) exploits use a specific attack sequence designed to take advantage of a vulnerability to gain access to a system; (3) DoS and DDoS attacks, during which the attacker tries to prevent the normal usage of a system, overload the system with requests so that the system's ability to process them efficiently is compromised or disrupted.[14]

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 signature-based method is that a slow, methodical attack might escape detection if the relevant IDPS attack signature has a shorter time frame. The only way a signature-based IDPS can resolve this vulnerability is to collect and analyze data over longer periods of time, a process that requires substantially larger data storage capability and additional processing capacity.

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 baseline data can include variables such as host memory or CPU usage, network packet types, and packet quantities.

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. If the actions of the users or systems on a network vary widely, with periods of low activity interspersed with periods of heavy packet traffic, this type of IDPS may not be suitable, because the dramatic swings from one level to another will almost certainly generate false alarms. Because of its complexity and impact on the overhead computing load of the host computer as well as the number of false positives it can generate, this type of IDPS is less commonly used than the signature-based type.

Stateful Protocol Analysis IDPS As you learned in Chapter 6, stateful inspection firewalls track each network connection between internal and external systems using a state table to record which station sent which packet and when, essentially pairing communicating parties. An IDPS extension of this concept is stateful protocol analysis. According to SP 800-94, "**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.

Stateful protocol analysis can also examine authentication sessions for suspicious activity as well as for attacks that incorporate "unexpected sequences of commands, such as issuing the same command repeatedly or issuing a command without first issuing a command upon which it is dependent, as well as 'reasonableness' for commands such as minimum and maximum lengths for arguments."[16]

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. Also, proprietary protocols are not published in sufficient detail to enable the IDPS to provide accurate and comprehensive assessments.

Unfortunately, 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.[17]

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. The patterns that signify an attack can be subtle and difficult to distinguish when one system is examined in isolation, but they may be more identifiable when the events recorded for the entire network and each of the systems in it can be viewed as a whole. Of course this holistic approach requires considerable resources since it involves the collection, movement, storage, and analysis of very large quantities of log data.

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. For example, if an NIDPS reacts to suspected DoS attacks by severing the network connection, the attack is a success, and such attacks repeated at intervals will thoroughly disrupt an organization's business operations.

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 passive IDPS is the most common implementation, although most systems include some active options that are disabled by default.

The following list describes some of the responses an IDPS can be configured to produce. Note that some of these apply only to a network-based or a host-based IDPS, while others are applicable to both.[18]

- 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, the tools used in the attack, the level of confidence the system has in its own determination, and the addresses and/or locations of the systems involved.
- 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, including IDPS traps, and the ability to use standard communications networks.
- 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—e-mail is inherently unreliable, and an attacker could compromise the e-mail system and block such messages.

- 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 systems outside that perimeter may make the organization just as criminally liable as the individual(s) who began the attack. Also, in some cases the "attacking system" is in fact a compromised intermediary system, and in other cases attackers use address spoofing; either way, any counterattack would actually only harm an innocent third party. Any organization planning to configure any sort of retaliation effort into an automated intrusion detection system is strongly encouraged to seek legal counsel.
- 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 suspected packets by IP address, port, or protocol. (It is, unfortunately, still possible for a skilled attacker to break in by simply spoofing a different address, shifting to a different port, or changing the protocols used in the attack.) 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 Internet connection) if the severity of the suspected attack warrants that level of response.[19]

- 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. As indicated earlier, this response should be the last resort to protect information, as it may be the very goal of the attacker.

[*The following sections have been adapted from NIST* SP 800-94 "Guide to Intrusion Detection and Prevention Systems" and its predecessor, *SP 800-31* "Intrusion Detection Systems".]

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?

  First, specify the technical attributes of your systems environment—network diagrams and maps specifying the number and locations of hosts; operating systems for each

host; the number and types of network devices such as routers, bridges, and switches; the number and types of terminal servers and dial-up connections; and descriptions of any network servers, including types, configurations, and application software and versions running on each. If you run an enterprise network management system, specify it here.

- What are the technical specifications of your current security protections?

  Describe the security protections you already have in place. Specify numbers, types, and locations of network firewalls, identification and authentication servers, data and link encryptors, antivirus packages, access control products, specialized security hardware (such as crypto accelerator hardware for Web servers), virtual private networks, and any other security mechanisms on your systems.

- What are the goals of your enterprise?

  Some IDPSs are designed to accommodate the special needs of certain industries or market niches such as electronic commerce, health care, or financial services. Define the functional goals of your enterprise (there can be several goals associated with a single organization) that are supported by your systems.

- How formal is the system environment and management culture in your organization?

  Organizational styles vary, depending on the function of the organization and its traditional culture. For instance, the military and other organizations that deal with national security issues tend to operate with a high degree of formality, especially when contrasted with university or other academic environments. Some IDPSs support enforcement of formal use policies, with built-in configuration options that can enforce common issue-specific or system-specific security policies, as well as provide a library of reports for typical policy violations as well as routine matters.

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?

  Perhaps the easiest way to identify security goals is by categorizing your organization's threat concerns. Identify the concerns that your organization has regarding external threats.

- Is your organization concerned about insider attack?

  Address concerns about threats that originate from within your organization, encompassing not only a user who attacks the system from within (such as a shipping clerk who attempts to access and alter the payroll system) but also the authorized user who exceeds his privileges, thereby violating organizational security policy or laws (such as a customer service agent who, driven by curiosity, accesses earnings and payroll records for public figures).

- Does your organization want to use the output of your IDPS to determine new needs?

  System usage monitoring is sometimes provided as a generic system management tool to determine when system assets require upgrading or replacement.

- Does your organization want to use an IDPS to maintain managerial control (non-security related) over network usage?

  Some organizations, implement system use policies that may be classified as personnel management rather than system security, such as prohibiting access to certain kinds of Web sites (such as ones containing pornography) or the use of organizational systems to send e-mail or other messages for the purpose of harassing individuals. Some IDPSs provide features that detect such violations of management controls.

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?

  It is helpful to articulate the goals outlined in the security policy in terms of the standard security goals (integrity, confidentiality, and availability) as well as more generic management goals (privacy, protection from liability, and manageability).

- What are the general job descriptions of your system users?

  List the general job functions of system users (there are often several functions assigned to a single user) as well as the data and network accesses that each function requires.

- Does the policy include reasonable use policies or other management provisions?

  As mentioned above, the security policies of many organizations include system use policies.

- Has your organization defined processes for dealing with specific policy violations?

  It is helpful to have a clear idea of what the organization wishes to do when the IDPS detects that a policy has been violated. If the organization doesn't intend to react to such violations, it may not make sense to configure the IDPS to detect them. If, on the other hand, the organization wishes to actively respond to such violations, the IDPS's operational staff should be informed of the response policy so that it can deal with alarms in an appropriate manner.

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?

  If so, does that oversight authority require IDPSs or other specific system security resources?

- Are there requirements for public access to information on your organization's systems?

  Do regulations or statutes require that information on your system be accessible by the public during certain hours of the day, or during certain date or time intervals?

- 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 hardware, software, and infrastructure?

  Remember that the IDPS software is not the only element of the total cost of ownership; you may also have to acquire a system on which to run the software, obtain specialized assistance to install and configure the system, and train your personnel. Ongoing operations may also require additional staff or outside contractors.
- Is there sufficient existing staff to monitor an intrusion detection system full time?

  Some IDPSs require around-the-clock attendance by systems personnel. If you do not anticipate having such personnel available, you may wish to explore those systems that accommodate less than full-time attendance or unattended use.
- Does your organization have authority to instigate changes based on the findings of an intrusion detection system?

  It is critical that you and your organization be clear about what you plan to do about the problems uncovered by an IDPS. If you are not empowered to handle the incidents that arise as a result of the monitoring, you should consider coordinating your selection and configuration of the IDPS with the party who is empowered.

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?

  Ask the vendor about the assumptions made regarding the goals and constraints of customer environments.

- Has the product been tested against attack?

  Ask vendors for details of the security testing to which its products have been subjected. If the product includes network-based vulnerability assessment features, ask also whether test routines that produce system crashes or other denials of service have been identified and flagged in system documentation and interfaces.

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?

  Ask here whether the IDPS's interface can be configured (with shortcut keys, customizable alarm features, and custom signatures) on the fly. Ask also whether these features are documented and supported.

- Can the product adapt to growth and change of the organization's systems infrastructure?

  This question has to do with the ability of the IDPS to scale to an expanding and increasingly diverse network. Most vendors have experience in adapting their products as target networks grow. Ask also about commitments to support new protocol standards and platform types.

- Can the product adapt to growth and change in the security threat environment?

  This question is especially critical given the current Internet threat environment, in which thirty to forty new attacks are posted to the Web every month.

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?

  Many vendors provide expert assistance to customers installing and configuring IDPSs; others expect that your own staff will handle these functions and provide only telephone or e-mail help desk functions.

- What are the commitments for ongoing product support?

  Ask about the vendor's commitment to supporting your use of their IDPS product.

- Are subscriptions to signature updates included?

  Most IDPSs are misuse-detectors, so the value of the product is only as good as the signature database against which events are analyzed. Most vendors provide subscriptions to signature updates for some period of time (a year is typical).

- How often are subscriptions updated?

  In today's threat environment, in which thirty to forty new attacks are published every month, this is a critical question.

- How quickly after a new attack is made public will the vendor ship a new signature?

  If you are using IDPSs to protect highly visible or heavily traveled Internet sites, it is especially critical that you receive the signatures for new attacks as soon as possible.

- Are software updates included?

  Most IDPSs are software products and therefore subject to bugs and revisions. Ask the vendor about software update and bug patch support, and determine to what extent they are included in the product you purchase.

- How quickly will software updates and patches be issued after a problem is reported to the vendor?

  As software bugs in IDPSs can allow attackers to nullify their protective effect, it is extremely important that problems be fixed, reliably and quickly.

- Are technical support services included? What is the cost?

  In this category, technical support services mean vendor assistance in tuning or adapting your IDPS to accommodate special needs, be they monitoring a custom or legacy system within your enterprise or reporting IDPS results in a custom protocol or format.

- What are the provisions for contacting technical support (e-mail, telephone, online chat, Web-based reporting)?

  The contact provisions will likely tell you whether these technical support services are accessible enough to support incident handling or other time-sensitive needs.

- Are there any guarantees associated with the IDPS?

  As with other software products, IDPSs traditionally have few guarantees associated with them; however, in an attempt to gain market share, some vendors are initiating guarantee programs.

- What training resources does the vendor provide?

  Once an IDPS is selected, installed, and configured, it must still be operated by your personnel. In order for these people to make optimal use of the IDPS, they should be trained in its use. Some vendors provide this training as part of the product package.

- What additional training resources are available from the vendor and at what cost?

  If the vendor does not provide training as part of the IDPS package, you should budget appropriately to train your operational personnel.

Strengths and Limitations of IDPSs

Although intrusion detection systems are a valuable addition to an organization's security infrastructure, there are things they do well and things they do not do well. As you plan the security strategy for your organization's systems, it is important for you to understand what IDPSs should be trusted to do and what goals might be better served by other security mechanisms.

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 behaviors
- 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

There is also the considerable challenge of configuring an IDPS to respond accurately to a perceived threat. Once a device is empowered to react to an intrusion by filtering or even severing an communication session or by severing a communication circuit, the impact from a false positive becomes significant. It's one thing to fill an administrator's e-mail box or compile a large log file with suspected attacks; it's quite another to shut down critical communications. 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.

[*The preceding sections were drawn and adapted from NIST SP 800-94* "Guide to Intrusion Detection and Prevention Systems" and its predecessor, *NIST SP 800-31* "Intrusion Detection Systems"]

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. The IT industry has been exploring technologies and practices to enable the distribution of computer processing cycles and data storage for many years. These explorations have long considered the advantages and disadvantages of the centralized strategy versus strategies with varying degrees of distribution. In the early days of computing, all systems were fully centralized, resulting in a control strategy that provided high levels of security and control, as well as efficiencies in resource allocation and management. During the 1980s and 1990s, with the rapid growth in networking and computing capabilities, the trend was to implement a fully distributed strategy. In the mid-1990s, however, the high costs of a fully distributed architecture became apparent, and the IT industry shifted toward a mixed strategy of partially distributed control. A strategy of partial distribution, where some features and components are distributed and others are centrally controlled, has now emerged as the recommended practice for IT systems in general and for IDPS control systems in particular.

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 (triangular 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. This centralization of IDPS management supports task specialization, since all managers are either located near the IDPS management console

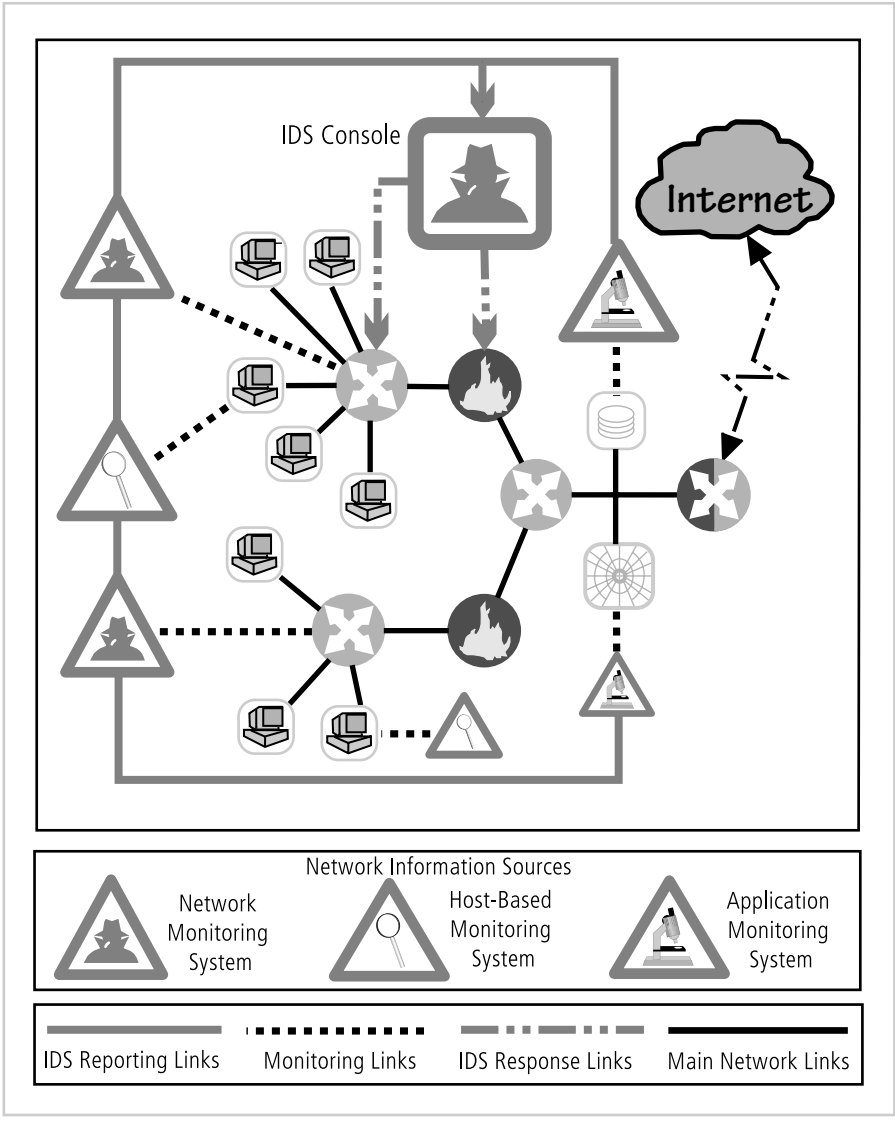

Figure 7-4 Centralized IDPS Control[13]

Source: This figure adapted from Scarfone and Mell, NIST SP800-94.

or can acquire an authenticated remote connection to it, and technicians are located near the remote sensors. This means that each person can focus specifically on an assigned task. In addition, the central control group can evaluate the systems and networks as a whole, and since it can compare pieces of information from all sensors, the group is better positioned to recognize a large-scale attack.

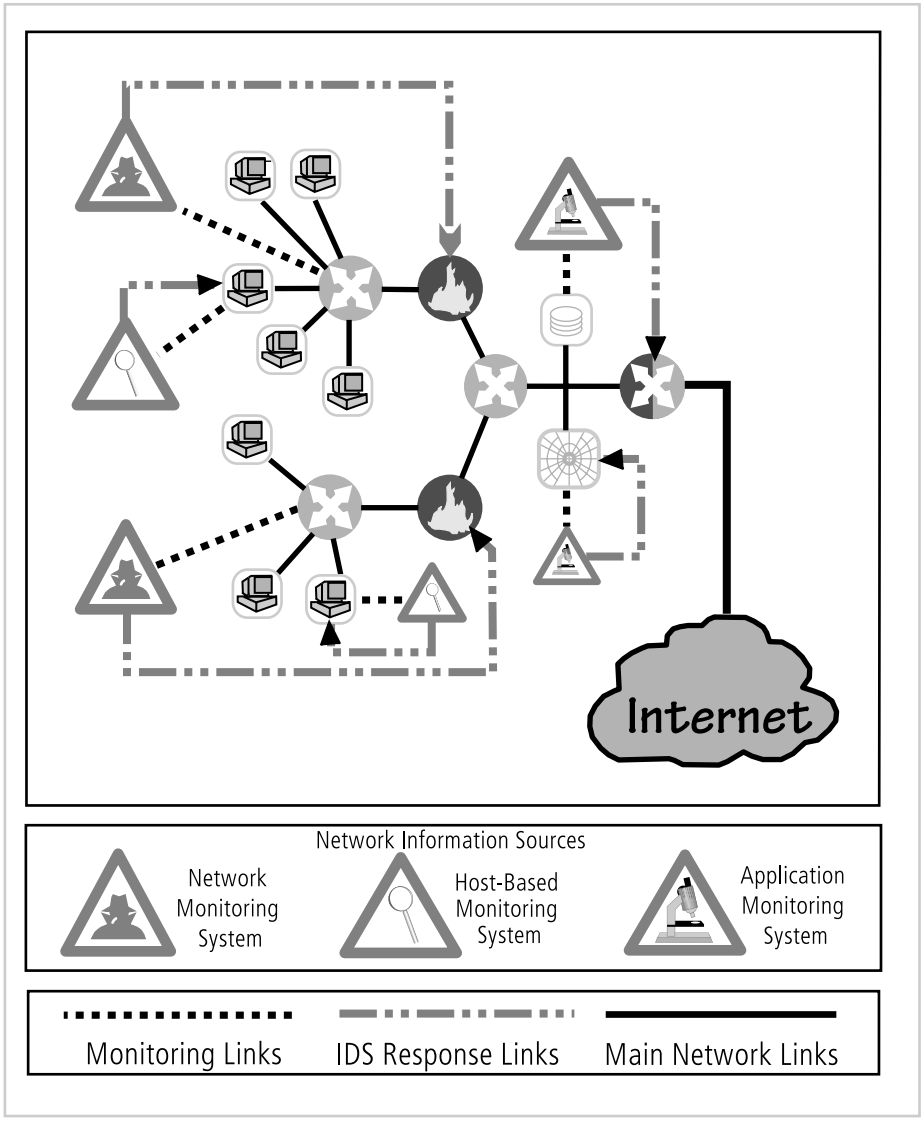

Figure 7-5 Fully Distributed IDPS Control[14]

Source: This figure adapted from Scarfone and Mell, NIST SP800-94.

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.

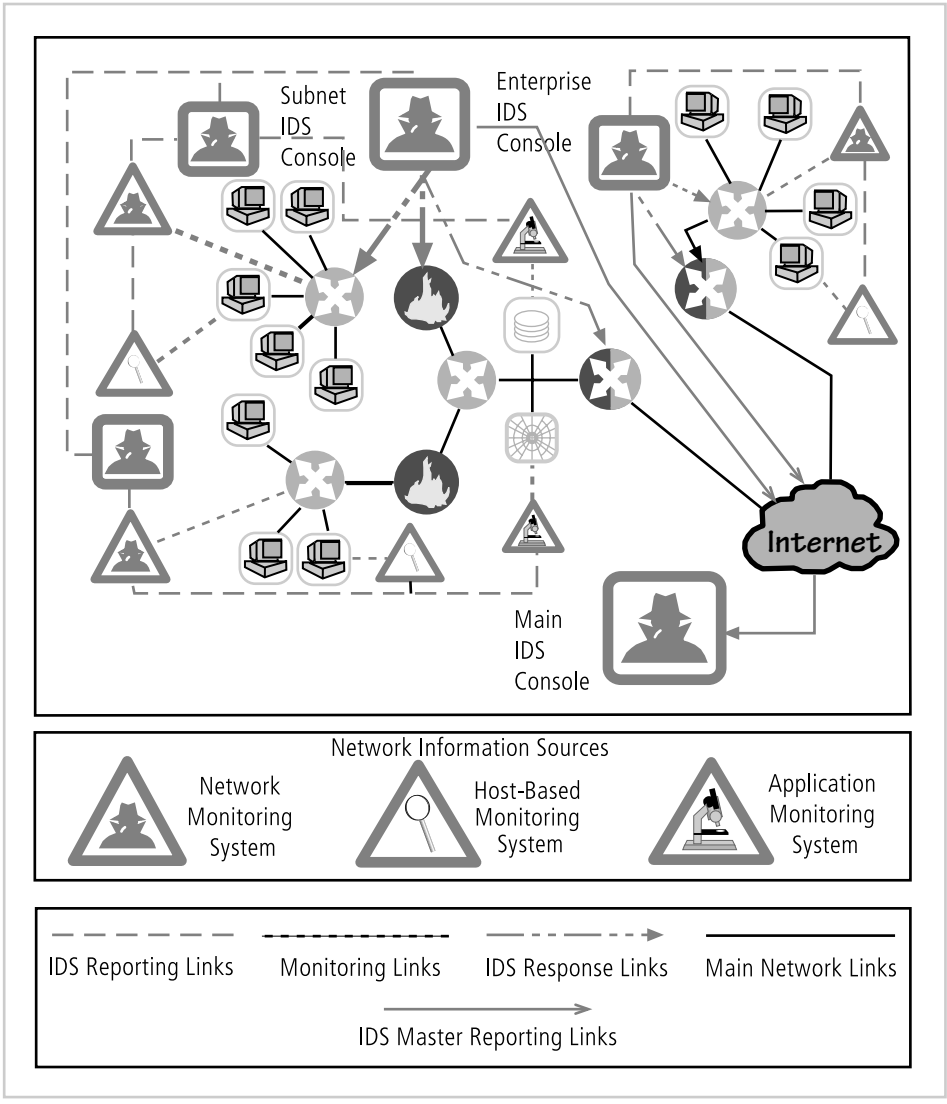

Figure 7-6 Partially Distributed IDPS Control[15]

Source: This figure adapted from Scarfone and Mell, NIST SP800-94.

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.

IDPS 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. Moreover, the administrators monitoring systems located in such areas can become desensitized to the information flow and may fail to detect actual attacks in progress.

As an organization selects an IDPS and prepares for implementation, planners must select a deployment strategy that is based on a careful analysis of the organization's information security requirements and that integrates with the organization's existing IT infrastructure but, at the same time, causes minimal impact. After all, the purpose of the IDPS is to detect anomalous situations—not create them. One consideration is the skill level of the personnel who install, configure, and maintain the systems. An IDPS is a complex system in that it involves numerous remote monitoring agents (on both individual systems and networks) that require proper configuration to gain the proper authentication and authorization. As the IDPS is deployed, each component should be installed, configured, fine-tuned, tested, and monitored. A mistake in any step of the deployment process may produce a range of problems—from a minor inconvenience to a network-wide disaster. Thus, both the individuals installing the IDPS and the individuals using and managing the system require proper training.

NIDPS and HIDPS can be used in tandem to cover both the individual systems that connect to an organization's networks and the networks themselves. To do this, it is important for an organization to use a phased implementation strategy so as not to affect the entire organization all at once. A phased implementation strategy also allows security technicians to resolve the problems that do arise without compromising the very information security the IDPS is installed to protect. When sequencing the implementation, the organization should first implement the NIDPSs, as they are less problematic and easier to configure than their host-based counterparts. After the NIDPSs are configured and running without issue, the HIDPSs can be installed to protect the critical systems on the host server. Once the NIDPSs and HIDPSs are both operational, administrators should scan the network with a vulnerability scanner like Nmap or Nessus to determine if (a) the scanners pick up anything new or unusual, and (b) if the IDPS can detect the scans.

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 (See Figure 7-7, location 1)

Advantages:

- IDPS sees attacks that originate from the outside that may penetrate the network's perimeter defenses.

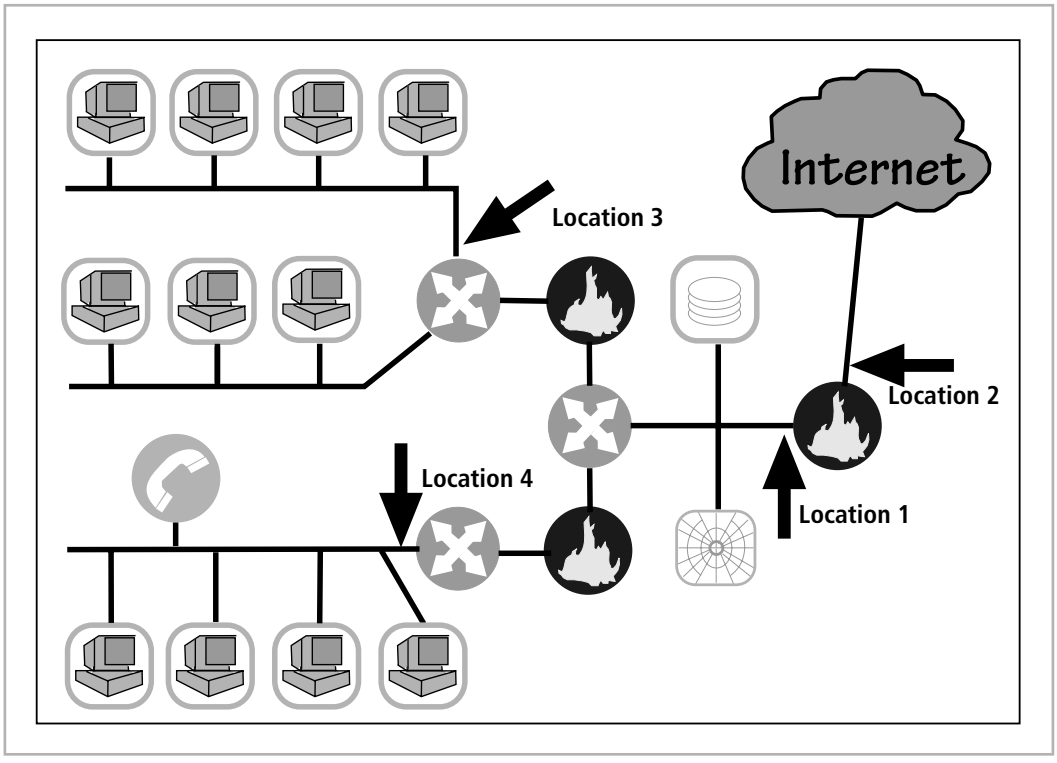

Figure 7-7 Network IDPS Sensor Locations[17]

Source: This figure adapted from Scarfone and Mell, NIST SP800-94.

- IDPS 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.
- Even if the incoming attack is not detected, the IDPS can sometimes recognize, in the outgoing traffic, patterns that suggest that the server has been compromised.

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 security 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.[20]

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. Practice helps the installation team gain experience and also helps determine if the installation might trigger any unusual events. Gaining an edge on the learning curve by training on nonproduction systems benefits the overall deployment process by reducing the risk of unforeseen complications.

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 percentage 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 configured 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. During the system testing process, training scenarios can be developed that will enable users to recognize and respond to common attack situations. To ensure effective and efficient operation, the management team can establish policy for the operation and monitoring of the HIDPS.

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. 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. Some IDPSs generate dynamic blacklists that are used to temporarily block recently detected threats (e.g., activity from an attacker's IP address). 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. Whitelists and blacklists are most commonly used in signature-based detection and stateful protocol analysis.

- *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*

  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.*[18]

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. This evaluation method of assessment was flawed for several reasons. Not all IDPSs use simple signature-based detection. Some systems, as discussed earlier, 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

This last measure is important, since future IDPSs will probably include much more detailed information about the overall site configuration. According to experts in the field, "it may be necessary for the IDPSs to be able to actively probe a potentially vulnerable machine, in order to either pre-load its configuration with correct information, or perform a retroactive assessment. An IDPS that performed some kind of actual system assessment would be a complete failure in today's generic testing labs, which focus on replaying attacks and scans against nonexistent machines."[19]

With the rapid growth in technology, each new generation of IDPSs will require new testing methodologies. However, the measured values that will continue to be of interest to IDPS administrators and managers will most certainly include some assessment of how much traffic the IDPS can handle, the numbers of false positives and false negatives it generates, and a measure of the IDPS's ability to detect actual attacks. Vendors of IDPS systems could also include a report of the alarms sent and the relative accuracy of the system in correctly matching the alarm level to the true seriousness of the threat. Some planned metrics for IDPSs include the flexibility of signatures and detection policy customization.

IDPS administrators may soon be able to purchase tools that test IDPS effectiveness. Until these tools are available from a neutral third party, the diagnostics from the IDPS vendors will always be suspect. No vendor, no matter how reliable, would provide a test that their system would fail.

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. In one reported example, a program was used to create a synthetic load of network traffic made up of many TCP sessions, with each session consisting of a SYN (or synchronization) packet, a series of data, and ACK (or acknowledgement) packets, but no FIN or connection termination packets. Of the several IDPS systems tested, one of them crashed due to lack of resources while it waited for the sessions to be closed. Another IDPS passed the test with flying colors because it did not perform state tracking on the connections. Neither of the tested IDPS systems worked as expected, but the one that didn't perform state tracking was able to stay operational and was, therefore, given a better score on the test.[20]

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. To understand why these tools are not yet widely used, you must first understand how they differ from a traditional IDPS. **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. This combination is meant to lure potential attackers into committing an attack, thereby revealing themselves—the idea being that once organizations have detected these attackers, they can better defend their networks against future attacks targeting real assets. 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

Because the information in a honeypot appears to be valuable, any unauthorized access to it constitutes suspicious activity. Honeypots are instrumented with sensitive monitors and event loggers that detect attempts to access the system and collect information about the potential attacker's activities. A screenshot from a simple IDPS that specializes in honeypot techniques, called Deception Toolkit, is shown in Figure 7-8. This screenshot shows the configuration of the honeypot as it is waiting for an attack.

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." As in honeypots, this environment can be filled with interesting data, which can convince an attacker that the attack is going according to plan. Like honeypots, padded cells are well-instrumented and offer unique opportunities for a target organization to monitor the actions of an attacker.

IDPS researchers have used padded cell and honeypot systems since the late 1980s, but until recently no commercial versions of these products were available. It is important to seek guidance from legal counsel before deciding to use either of these systems in your operational environment, since using an attractant and then launching a back hack or counterstrike might be illegal, and could make the organization vulnerable to a lawsuit or criminal complaint.

The advantages and disadvantages of using the honeypot or padded cell approach are summarized below:

Advantages:

- Attackers can be diverted to targets that they cannot damage.
- Administrators have time to decide how to respond to an attacker.

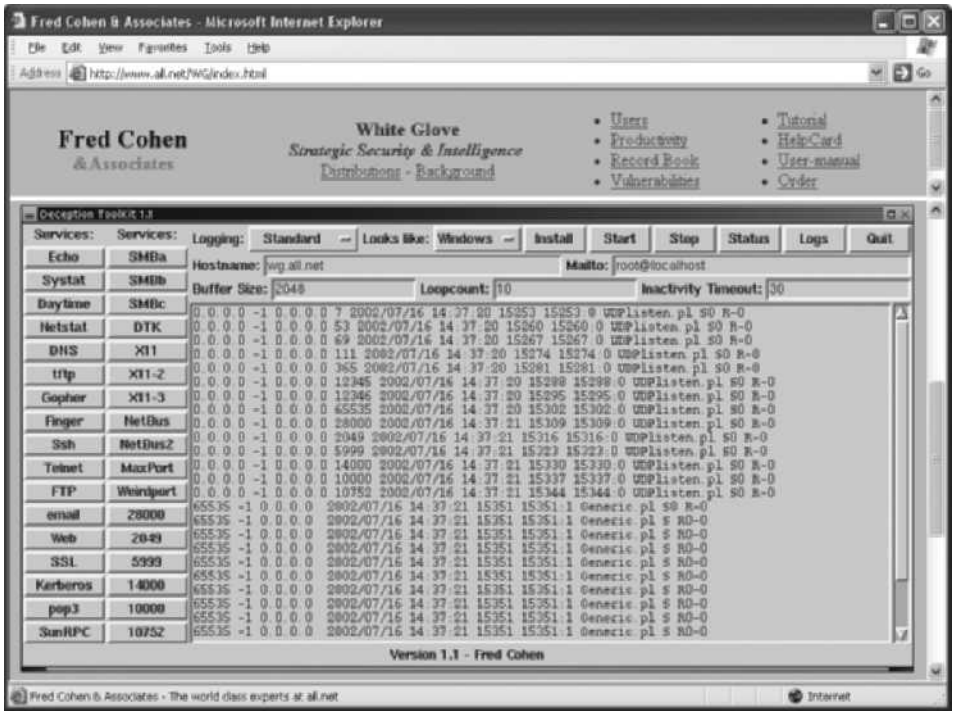

Figure 7-8 Deception Toolkit

Source: Course Technology/Cengage Learning

- 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.[21]

Trap-and-Trace Systems

Trap-and-trace applications, which are an extension of the attractant technologies discussed in the previous section, are growing in popularity. 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. One popular professional trap-and-trace software suite, ManHunt, and its companion honeypot application, ManTrap, was discontinued in 2006. No similar products have arisen to take their place, due to the drawbacks and complications of using these technologies.

On the surface, 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. Vigilante justice would be a more appropriate term for these activities, which are in fact deemed unethical by most codes of professional conduct. In tracking the hacker, administrators may end up wandering through other organizations' systems, especially when the wily hacker has used IP spoofing, compromised systems, or a myriad of other techniques to throw trackers off the trail. The backhacking administrator becomes 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. It is difficult to gauge the effect such a system can have on the average user, especially if the individual has been nudged into looking at the information. 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 (http://*labrea.sourceforge.net/labrea-info.html*). 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. They install a simple perimeter firewall, and then, lulled into a sense of security by this single layer of defense, they relax. 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). As you've learned, the first item in this list, 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. Many of the best scanning and analysis tools are those developed by the hacker community and are available free on the Web. Good administrators should have several hacking Web sites bookmarked and should try to keep up with chat room discussions on new vulnerabilities, recent conquests, and favorite assault techniques. There is nothing wrong with a security administrator using the tools that potential attackers use in order to examine network defenses and find areas that require additional attention. 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. A word of caution, though: many of these scanning and analysis tools have distinct signatures, and some Internet service providers (ISPs) scan for these signatures. If the ISP discovers someone using hacker tools, it can pull that person's access privileges. It is probably best for administrators to establish a working relationship with their ISPs and notify the ISP of their plans.

Scanning tools are, as mentioned earlier, typically 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. **Footprinting** is the organized research of the Internet addresses owned or controlled by a target organization. The attacker uses public Internet data sources to perform keyword searches to identify the network addresses of the organization. This research is augmented by browsing the organization's Web pages. 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.

In addition, public business Web sites (such as Forbes or Yahoo Business) often reveal information about company structure, commonly used company names, and other information that attackers find useful. Furthermore, common search engines allow attackers to query for any site that links to their proposed target. By doing a little bit of initial Internet research into a company, an attacker can often find additional Internet locations that are not commonly associated with the company—that is, business-to-business (B2B) partners and subsidiaries. Armed with this information, the attacker can find the "weakest link" into the target network.

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. While it may seem trite or clichéd, the phrase "a chain is only as strong as its weakest link" is very relevant to network and computer security. If a company has a trusted network connection with fifteen business partners, one weak business partner can compromise all sixteen networks.

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. Since the original site no longer offers the software, to obtain it you must search the Web for a copy of the last version (1.14). A sample screenshot from Sam Spade is shown in Figure 7-9. 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 (known, from the commonly used UNIX command for performing the analysis, as traceroutes). All of these are powerful diagnostic and hacking activities. 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.

For Linux or BSD systems, there is a tool called "wget" that allows a remote individual to "mirror" entire Web sites. With this tool, attackers can copy an entire Web site and then go through the source HTML, JavaScript, and Web-based forms at their leisure, collecting and collating all of the data from the source code that will be useful to them for their attack.

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. Fingerprinting, which deploys various tools as described in the following sections, reveals useful information about the internal structure and operational nature of the target system or network for the anticipated attack. Since these tools were created to find vulnerabilities in systems and networks quickly and with a minimum of effort, they are valuable to the network defender since they can quickly pinpoint the parts of the systems or network that need a prompt repair to close the vulnerability.

Figure 7-9 Sam Spade

Source: Course Technology/Cengage Learning

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. For instance, if you are trying to identify a Windows computer in a typical network, a built-in feature of the operating system, nbtstat, may be able to get the answer you need very quickly without the use of a scanner. This tool will not work on other types of networks, however, so you must know your tools in order to make the best use of the features of each.

The more specific the scanner is, the more useful the information it provides to attackers and defenders. However, you should keep a generic, broad-based scanner in your toolbox to help locate and identify rogue nodes on the network that administrators may be unaware of. Probably the most popular port scanner is Nmap, which runs on both Unix and Windows systems. You can find out more about Nmap at *www.insecure.org*.

A port is a network channel or connection point in a data communications system. Within the TCP/IP networking protocol, TCP and User Datagram Protocol (UDP) port numbers differentiate the multiple communication channels that are used to connect to the network services being offered on the same network device. Each application within TCP/IP has a unique port number. Some have default ports but can also use other ports. Some of the well-known port numbers are presented in Table 7-1. In all, there are 65,536 port numbers in use for TCP and another 65,536 port numbers for UDP. Services using the TCP/IP protocol can run

TCP Port Numbers	TCP Service
20 and 21	File Transfer Protocol (FTP)
22	Secure Shell (SSH)
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name Services (DNS)
67 and 68	Dynamic Host Configuration Protocol (DHCP)
80	Hypertext Transfer Protocol (HTTP)
110	Post Office Protocol (POP3)
161	Simple Network Management Protocol (SNMP)
194	IRC chat port (used for device sharing)
443	HTTP over SSL
8080	Used for proxy services

Table 7-1 Select Commonly Used Port Numbers

on any port; however, services with reserved ports generally run on ports 1–1023. Port 0 is not used. Ports greater than 1023 are typically referred to as ephemeral ports and may be randomly allocated to server and client processes.

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. More information on Firewalk can be obtained from *www.packetstormsecurity.org/UNIX/audit/firewalk*.

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. HPING can be found at *www.hping.org*.

Incidentally, administrators who are wary of using the same tools that attackers use should remember two important points: regardless of the tool that is used to validate or analyze a firewall's configuration, it is user intent that dictates how the information gathered is used; in order to defend a computer or network well, it is necessary to understand the ways it can be attacked. Thus, a tool that can help close up an open or poorly configured firewall will help the network defender minimize the risk from attack.

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. This tool can be found at *www.sourceforge.net/projects/xprobe*. 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. System and network administrators should take note of this and restrict the use of ICMP through their organization's firewalls and, when possible, within its internal networks.

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. Figures 7-10 and 7-11 show sample LANguard and Nessus result screens.

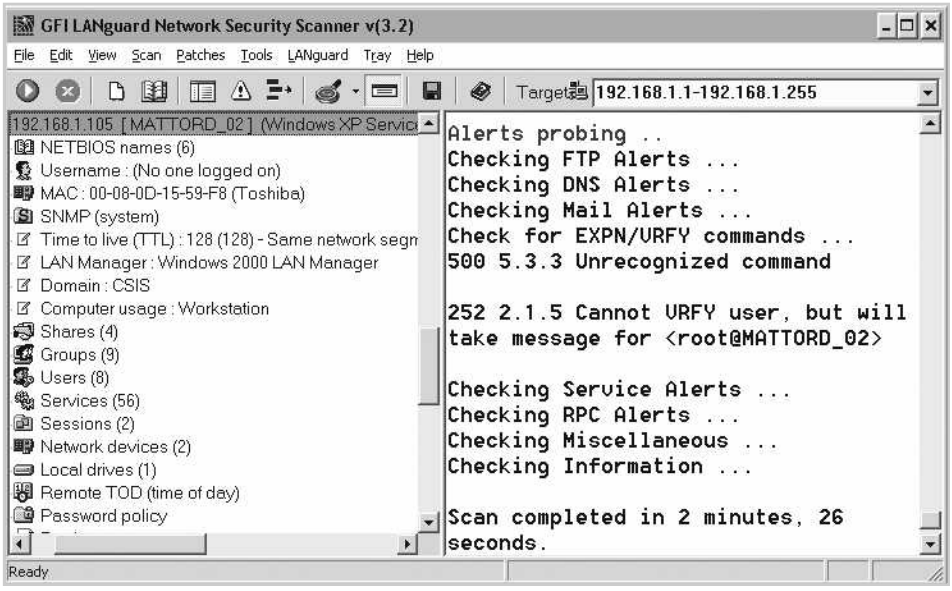

Figure 7-10 LANguard

Source: Course Technology/Cengage Learning

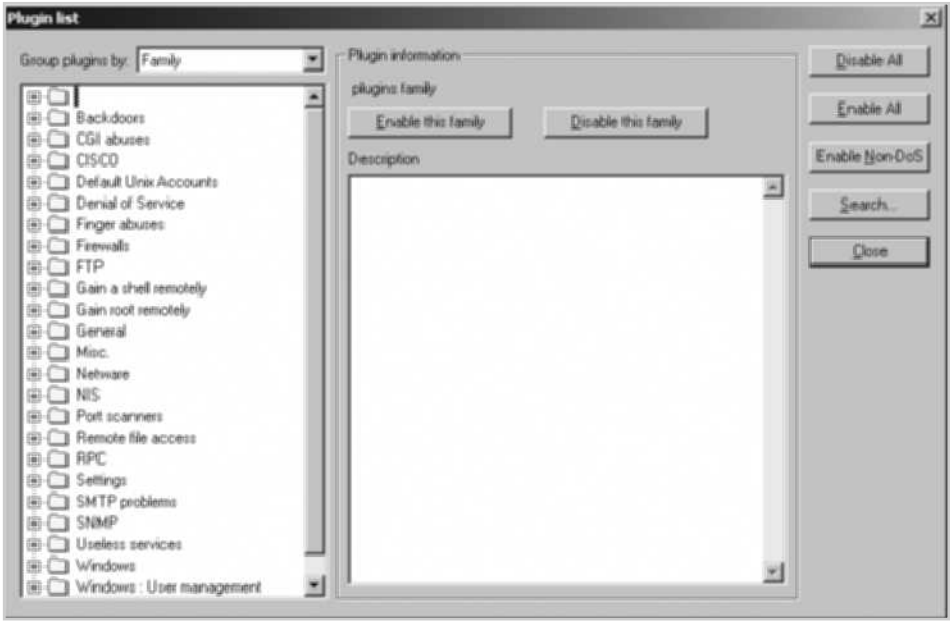

Figure 7-11 Nessus

Source: Course Technology/Cengage Learning

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. One example of a fuzz scanner is SPIKE, which has two primary components. The first is the SPIKE Proxy, which is a full-blown proxy server. As Web site visitors utilize the proxy, SPIKE builds a database of each of the traversed pages, forms, and other Web-specific information. When the Web site owner determines that enough history has been collected to fully characterize the Web sites, SPIKE can be used to check the Web site for bugs—that is, administrators can use the usage history collected by SPIKE to traverse all known pages, forms, active programs (e.g., asp, cgi-bin), and so forth, and can test the system by attempting overflows, SQL injection, cross-site scripting, and many other classes of Web attacks.

SPIKE also has a core functionality to fuzz any protocol that utilizes TCP/IP. By sniffing a session and building a SPIKE script, or building a full-blown C program using the SPIKE API, a user can simulate and "fuzz" nearly any protocol. Figure 7-12 shows the SPIKE Proxy configuration screen. Figure 7-13 shows a sample SPIKE script being prepared to fuzz the ISAKAMP protocol (which is used by VPNs). Figure 7-14 shows the SPIKE program, generic_send_udp, fuzzing an IKE server using the SPIKE script. As you can see, SPIKE can be used to quickly fuzz and find weaknesses in nearly any protocol.

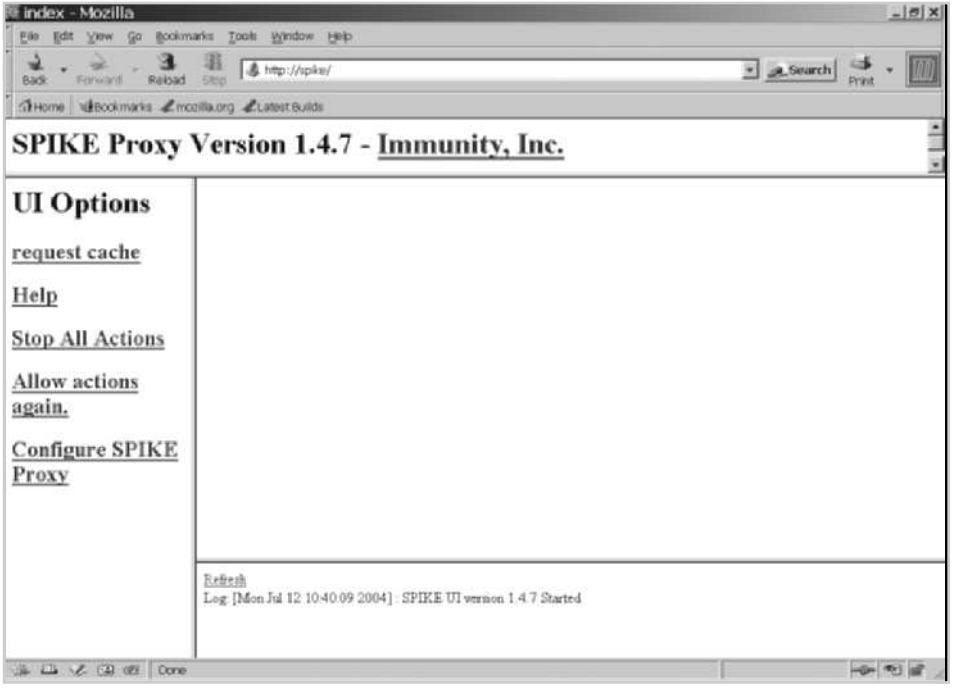

Figure 7-12 SPIKE Proxy

Source: Course Technology/Cengage Learning

Figure 7-13 SPIKE in Action

Source: Course Technology/Cengage Learning

Figure 7-14 SPIKE Versus IKE

Source: Course Technology/Cengage Learning

Similar in function, the 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. In fact, these tools are so powerful that even system defenders who use them are not likely to use them in the most aggressive modes on their production networks. At the time of this writing, the most popular scanners seem to be Nessus (a commercial version of Nessus for Windows is available), Retina, and Internet Scanner. The Nessus scanner is available at no cost; the other two require a license fee.

Often times, some members of an organization require proof that a system is actually vulnerable to a certain attack. They may require such proof in order to avoid having system administrators attempt to repair systems that are not in fact broken, or because they have not yet built a satisfactory relationship with the vulnerability assessment team. In these instances, there exists a class of scanners that actually exploit the remote machine and allow the vulnerability analyst (sometimes called a penetration tester) to create an account, modify a Web page, or view data. These tools can be very dangerous and should only be used when absolutely necessary. Three tools that can perform this action are Core Impact, Immunity's CANVAS, and the Metasploit Framework.

Of these three tools, only the Metasploit Framework is available without a license fee (see *www.metasploit.com*). The Metasploit Framework is a collection of exploits coupled with an interface that allows the penetration tester to automate the custom exploitation of vulnerable systems. For instance, if you wished to exploit a Microsoft Exchange server and run a single command (perhaps add the user "security" into the administrators group), the tool allows you to customize the overflow in this manner. See Figure 7-15 for a screenshot of the Metasploit Framework in action.

A **passive vulnerability scanner** is one that listens in on the network and determines vulnerable versions of both server and client software. At the time of this writing, there are two primary vendors offering this type of scanning solution: Tenable Network Security with its Passive Vulnerability Scanner (PVS) and Sourcefire with its RNA product. 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. For instance, an active scanner operating without DOMAIN Admin rights would be unable to determine the version of Internet Explorer running on a desktop machine, whereas a passive scanner can make that determination by observing the traffic to and from the client. See Figure 7-16 for a screenshot of the Tenable PVS passive vulnerability scanner running on Windows XP.

Table 7-2 provides Web addresses for the products mentioned in the vulnerability scanners section.

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. In the wrong hands, however, a sniffer can be used to eavesdrop on network traffic. There are both commercial and open-source sniffers—more

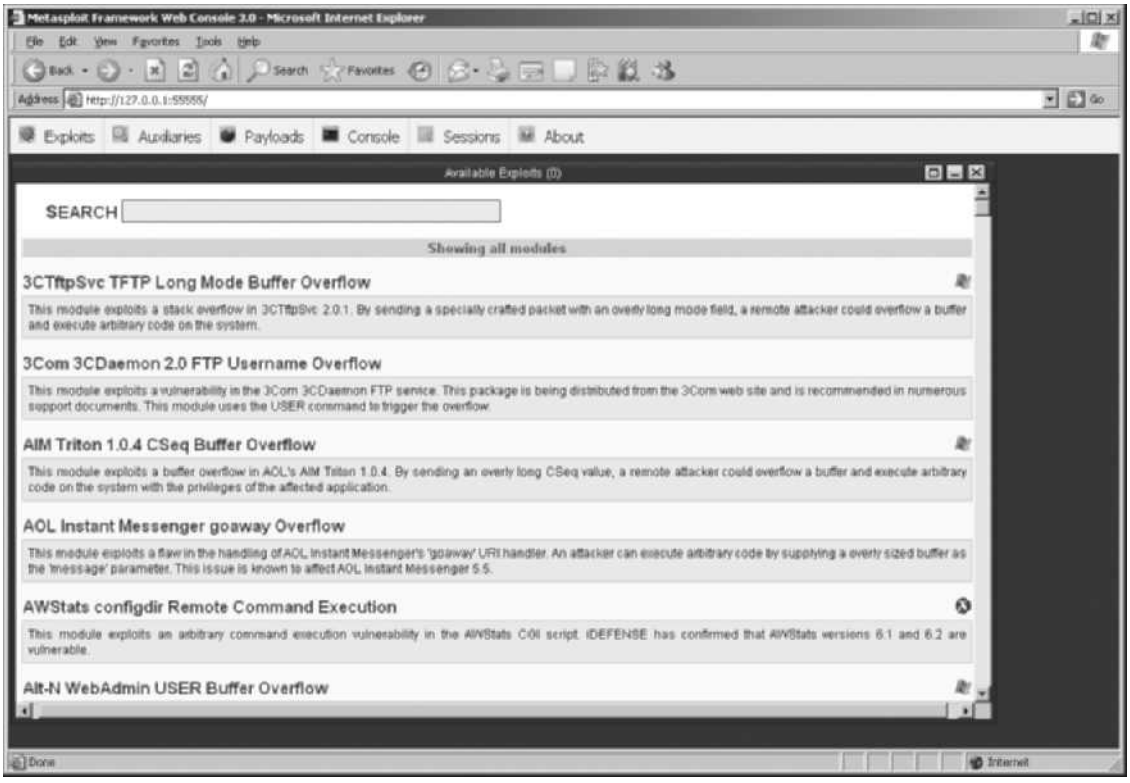

Figure 7-15 Metasploit

Source: Course Technology/Cengage Learning

specifically, Sniffer is a commercial product, and Snort is open-source software. An excellent free, client-based network protocol analyzer is Wireshark (*www.wireshark.org*), 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. Figure 7-17 shows a sample screen from Wireshark. 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. Incidentally, these three items are the same requirements for employee monitoring in general, and packet sniffing should be construed as a form of employee monitoring.

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

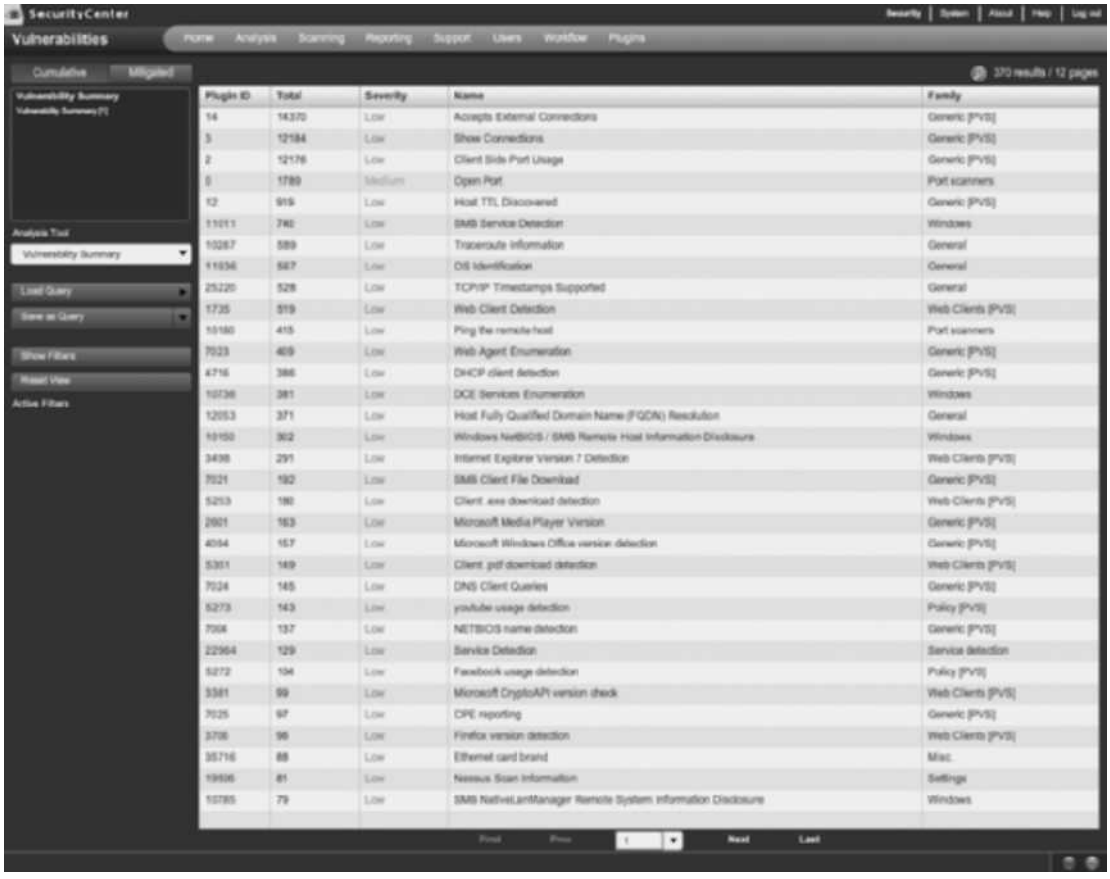

Figure 7-16 Tenable PVS

Source: Course Technology/Cengage Learning

Product	Web Page
Nessus	*www.nessus.org*
Nessus for Windows	*www.tenablesecurity.com*
GFI LANguard Network Security Scanner	*www.gfi.com/languard*
SPIKE – SPIKE Proxy	*www.immunitysec.com*
Retina	*www.eeye.com*
Internet Scanner	*www.iss.net*
Core Impact	*www.coresecurity.com*
Metasploit Framework	*www.metasploit.com*

Table 7-2 Vulnerability Scanner Products and Web Pages

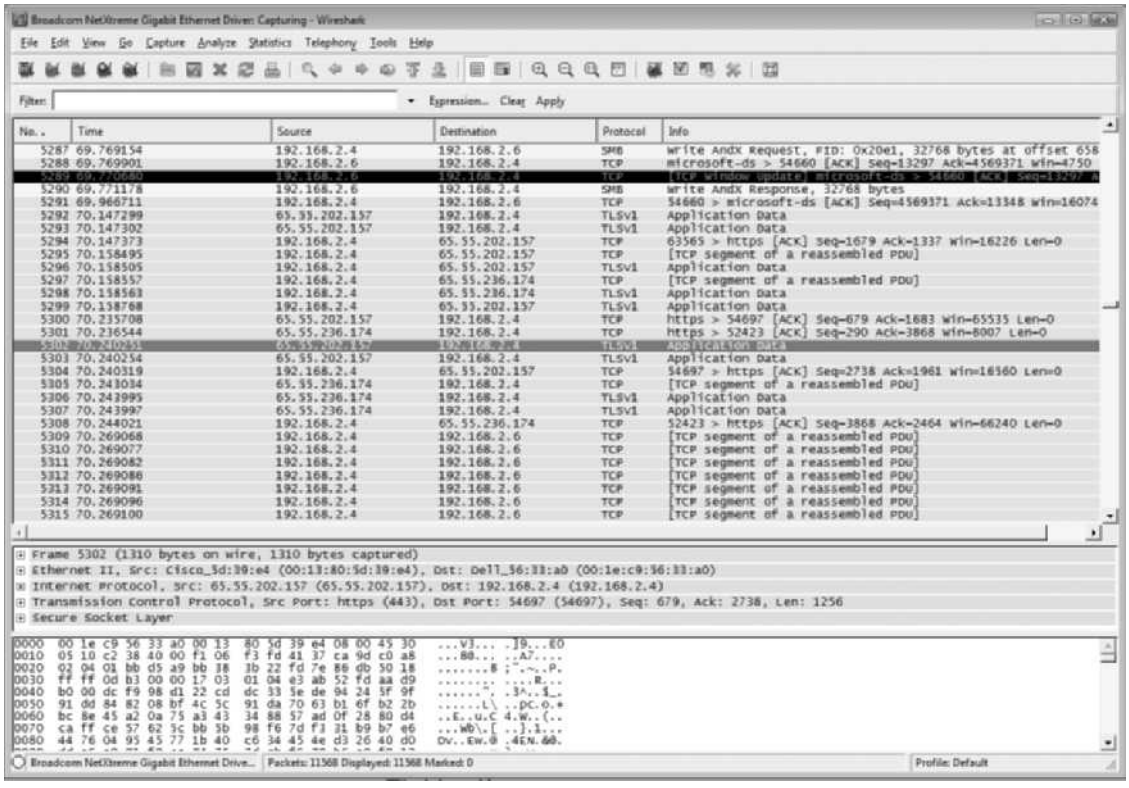

Figure 7-17 Wireshark

Source: Course Technology/Cengage Learning

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). To secure data in transit across any network, organizations must use encryption to be assured of content privacy.

Wireless Security Tools

802.11 wireless networks have sprung up as subnets on nearly all large networks. 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. In 2006, Insecure.org conducted a survey to identify the top five wireless tools. (See http://sectools.org/wireless.html) The winners were:

- Kismet, a powerful wireless sniffer, network detector, and IDPS, which works by passively sniffing the networks
- Netstumbler, a freeware Windows destumbler available at *www.netstumbler.org*
- Aircrack, a WEP/WPA cracking tool

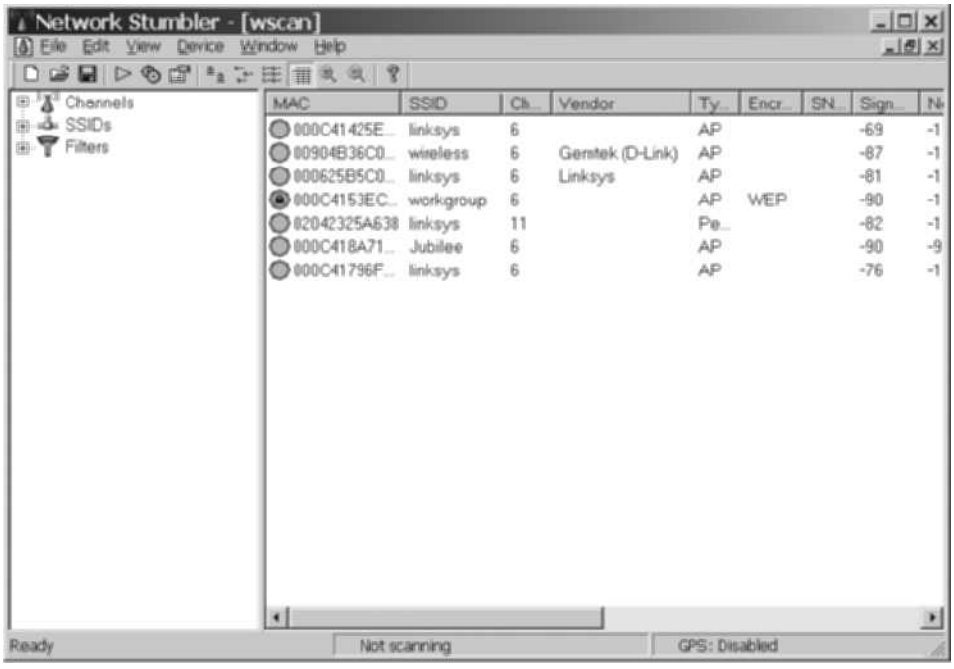

Figure 7-18 NetStumbler

Source: Course Technology/Cengage Learning

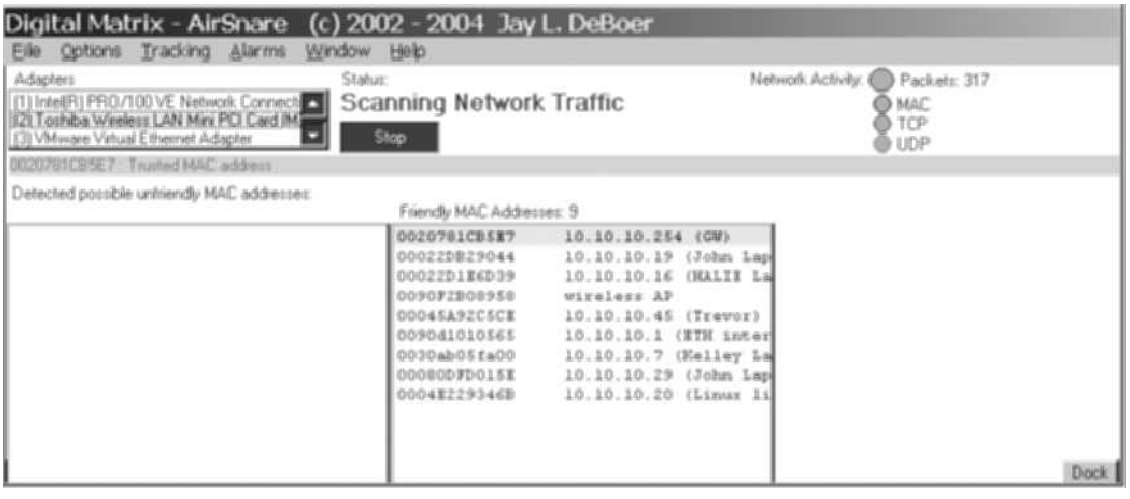

Figure 7-19 AirSnare

Source: Course Technology/Cengage Learning

- 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 and can be found at *www.netstumbler.org*. Figure 7-18 shows NetStumbler being run from a Windows XP machine. Another wireless tool worth

mentioning is AirSnare. 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. Figure 7-19 shows AirSnare in action.

The tools discussed so far help the attacker and the defender prepare themselves to complete the next steps in the attack protocol: attack, compromise, and exploit. These steps are beyond the scope of this text, and are usually covered in more advanced classes on computer and network attack and defense.

Biometric Access Controls

You learned the basics of access control and authentication in Chapter 6. In this section you will build on that foundation and learn about the technology associated with 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)

Figure 7-20 depicts some of these human recognition characteristics.

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. Each subsequent

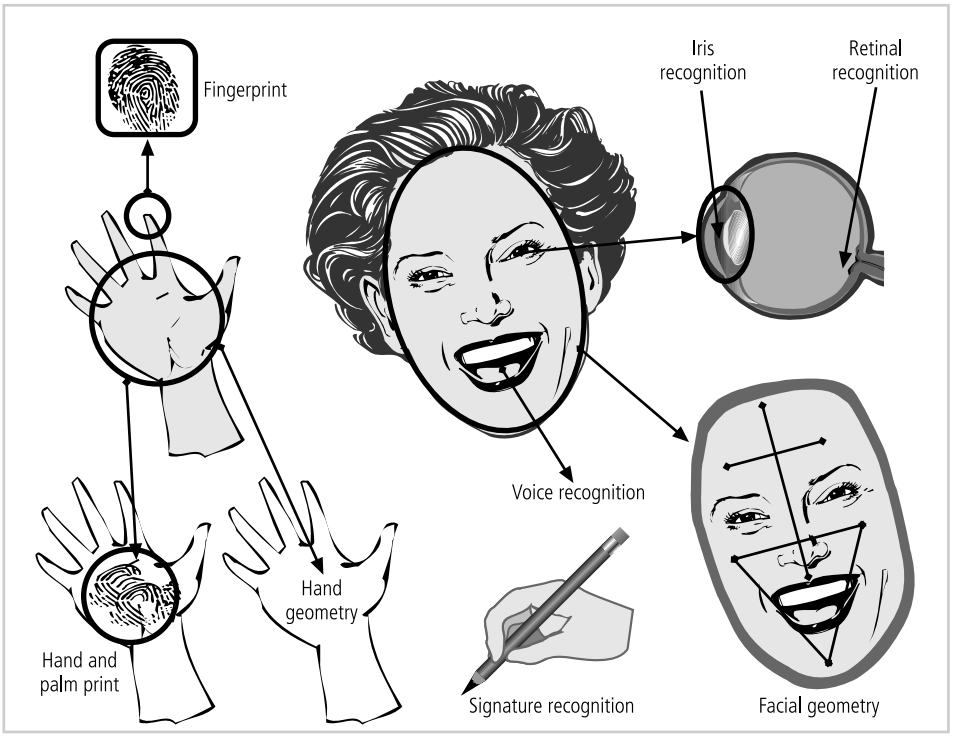

Figure 7-20 Biometric Recognition Characteristics

Source: Course Technology/Cengage Learning

access attempt results in a measurement that is compared with the encoded value to determine if the user is who he or she claims to be. 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. Currently, the technology for signature capturing is much more widely accepted than that for signature comparison, because signatures change due to a number of factors, including age, fatigue, and the speed with which the signature is written.

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. While a nuisance to supplicants who are authorized users, this error rate is probably of least concern to security professionals since rejection of an authorized user represents no threat to security. The false reject rate is often ignored unless it reaches a level high enough to generate complaints from irritated supplicants. 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. CERs are used to compare various biometrics and may vary by manufacturer. A biometric device that provides a CER of 1 percent is a device for which the failure rate for false rejection and the failure rate for false acceptance are both 1 percent. A device with a CER of 1 percent is considered superior to a device with a CER of 5 percent.

Acceptability of Biometrics

As you've learned, a balance must be struck between how acceptable a security system is to its users and how effective it is in maintaining security. 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. Table 7-3 shows how certain biometrics rank in terms of effectiveness and acceptance. Interestingly, the order of effectiveness is nearly exactly opposite the order of acceptance.

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand Vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

Table 7-3 Ranking of Biometric Effectiveness and Acceptance

H=High, M=Medium, L=Low

Reproduced from The '123' of Biometric Technology, *2003, by Yun, Yau Wei*[22]

Selected Readings

- *Intrusion Detection and Prevention* by Carl Endorf, Gene Schultz, and Jim Mellander. 2003, McGraw-Hill Osborne Media.
- *Guide to Biometrics* by Ruud Bolle, Jonathan Connell, Sharanthchandra Pankanti, Nalini Ratha, and Andrew Senior. 2003, Springer Professional Computing.
- National Institute of Standards and Technology (NIST) Special Publication 800-31, "Intrusion Detection Systems" by Rebecca Bace and Peter Mell. Available from the archive section of the NIST Computer Security Resource Center at *http://csrc.nist.gov*.
- National Institute of Standards and Technology (NIST) Special Publication 800-94, "Guide to Intrusion Detection and Prevention Systems" by Karen Scarfone and Peter Mell. Available from the NIST Computer Security Resource Center at *http://csrc.nist.gov*.

Chapter Summary

- Intrusion detection systems (IDSs) detect potential intrusions and sound an alarm. The more recently developed intrusion prevention systems (IPSs) also detect intrusions and can also take action to defend the network.
- An intrusion detection and prevention system (IDPS) works like a burglar alarm by detecting network traffic that is a violation of the rules with which it is configured (corresponding to an opened or broken window) and activates an alarm.

- A network-based IDPS (NIDPS) monitors network traffic, and when a predefined event occurs, it responds and notifies the appropriate administrator. A host-based IDPS (HIDPS) resides on a particular computer or server and monitors activity on that system.
- Signature-based IDPSs, also known as knowledge-based IDPSs, examine data traffic for patterns that match signatures, which are preconfigured, predetermined attack patterns. Statistical anomaly-based IDPSs, also known as behavior-based IDPSs, collect data from normal traffic and establish a baseline. When an activity is found to be outside the baseline parameters (or clipping level), these IDPSs activate an alarm to notify the administrator.
- Selecting IDPS products that best fit an organization's specific needs is a challenging and complex process since there are a wide array of products and vendors, each with its own approach and capabilities.
- Deploying and implementing IDPS technology is a complex undertaking that requires knowledge of the system and experience with the technology. After deployment, each organization should measure the effectiveness of its IDPS and then continue to assess its effectiveness periodically after the initial deployment.
- Honeypots are decoy systems designed to lure potential attackers away from critical systems. In the security industry, these systems are also known as decoys, lures, or flytraps. Two variations on this technology are known as honeynets and padded cell systems.
- Trap-and-trace applications are designed to react to an intrusion event by tracing it back to its source. This process is fraught with professional and ethical issues—some in the field believe that the back hack in the trace process is as significant a violation as the initial attack.
- Active intrusion prevention seeks to limit the damage that attackers can perpetrate by making the local network resistant to inappropriate use.
- Scanning and analysis tools are used to pinpoint vulnerabilities in systems, holes in security components, and unsecured aspects of the network. Although these tools are used by attackers, they can also be used by an administrator not only to learn more about his or her own system but also to identify and repair system weaknesses before they result in losses.
- Biometric authentication encompasses a set of technical means that measure one or more physical characteristics in order to verify a person's identity.
- Biometric technologies are evaluated on three basic criteria: the false reject rate, the false accept rate, and the crossover error rate.

Review Questions

1. What common security system is an IDPS most like? In what ways are these systems similar?
2. How does a false positive alarm differ from a false negative one? From a security perspective, which is least desirable?
3. How does a network-based IDPS differ from a host-based IDPS?

4. How does a signature-based IDPS differ from a behavior-based IDPS?
5. What is a monitoring (or SPAN) port? What is it used for?
6. List and describe the three control strategies proposed for IDPS control.
7. What is a honeypot? How is it different from a honeynet?
8. How does a padded cell system differ from a honeypot?
9. What is network footprinting? What is network fingerprinting? How are they related?
10. Why do many organizations ban port scanning activities on their internal networks? Why would ISPs ban outbound port scanning by their customers?
11. What is an open port? Why is it important to limit the number of open ports to only those that are absolutely essential?
12. What is a vulnerability scanner? How is it used to improve security?
13. What is the difference between active and passive vulnerability scanners?
14. What kind of data and information can be found using a packet sniffer?
15. What capabilities should a wireless security toolkit include?
16. What is biometric authentication? What does the term *biometric* mean?
17. Are any biometric recognition characteristics considered more reliable than others? Which are the most reliable?
18. What is a false reject rate? What is a false accept rate? What is their relationship to the crossover error rate?
19. What is the most widely accepted biometric authorization technology? Why do you think this technology is acceptable to users?
20. What is the most effective biometric authorization technology? Why do you think this technology is deemed to be most effective by security professionals?

Exercises

1. A key feature of hybrid IDPS systems is event correlation. After researching event correlation online, define the following terms as they are used in this process: compression, suppression, and generalization.
2. ZoneAlarm is a PC-based firewall and IDPS tool. Visit the product manufacturer at *www.zonelabs.com*, and find the product specification for the IDPS features of ZoneAlarm. Which of the ZoneAlarm products offer these features?
3. Using the Internet, search for commercial IDPS systems. What classification systems and descriptions are used, and how can these be used to compare the features and components of each IDPS? Create a comparison spreadsheet identifying the classification systems you find.
4. Use the Internet to find vendors of thumbprint and iris scanning tools. Which of these tools is more economical? Which of these is least intrusive?
5. There are several online passphrase generators available. Locate at least two of them on the Internet, and try them out. What did you observe?

Case Exercises

Miller Harrison was still working his way down his attack protocol.

Nmap started out as it usually did: giving the program identification and version number. Then it started reporting back on the first host in the SLS network. It reported all of the open ports on this server. Then the program moved on to a second host and began reporting back the open ports on that system, too. Once it reached the third host, however, it suddenly stopped.

Miller restarted Nmap, using the last host IP as the starting point for the next scan. No response. He opened up another command window and tried to ping the first host he had just port-scanned. No luck. He tried to ping the SLS firewall. Nothing. He happened to know the IP address for the SLS edge router. He pinged that and got the same result. He had been blackholed—meaning his IP address had been put on a list of addresses from which the SLS edge router would no longer accept packets. This was, ironically, his own doing. The IDPS he had been helping SLS configure seemed to be working just fine at the moment. His attempt to hack the SLS network was shut down cold.

Questions:

1. Do you think Miller is out of options as he pursues his vendetta? If you think there are additional actions he could take in his effort to damage the SLS network, what are they?
2. Suppose a system administrator at SLS happened to read the details of this case. What steps should he or she take to improve the company's information security program?

Endnotes

1. Scarfone, K., and Mell, P. "Guide to Intrusion Detection and Prevention Systems (IDPS)." NIST Special Publication 800-94. 2007 Accessed 21 June 2007 from http://*csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf*.
2. ibid.
3. ibid.
4. ibid.
5. ibid.
6. ibid.
7. ibid.
8. ibid.
9. ibid.
10. ibid.
11. Graham, R. "FAQ: Intrusion Detection Systems." March 2000. Accessed 21 June 2007 from *www.linuxsecurity.com/resource_files/intrusion_detection/network-intrusion-detection.html*.

12. Scarfone and Mell.
13. ibid.
14. ibid.
15. ibid.
16. ibid.
17. ibid.
18. ibid.
19. Ranum, Marcus J. "False Positives: A User's Guide to Making Sense of IDS Alarms," ICSA Labs IDSC. February 2003. Accessed 15 March 2004 from *www.icsalabs.com/html/communities/ids/whitepaper/FalsePositives.pdf*.
20. Scarfone and Mell.
21. ibid.
22. Yun, W. "The '123' of Biometric Technology." 2003. Accessed 21 November 2005 from *www.itsc.org.sg/synthesis/2002/biometric.pdf*.