Lesson Proper for Week 11

INTRUSION DETECTION AND PREVENTION SYSTEMS

Goals of an IDPS

A network intrusion is an attempt to gain unauthorized access to network resources. The term intrusion is a polite way of referring to an attack, which is often launched with the intention of compromising the integrity and confidentiality of network data or users' privacy. An intrusion detection and prevention system (IDPS) consists of more than one application or hardware device and incorporates more than just detection. Intrusion detection and prevention involve three network defense functions—prevention, detection, and response. As shown in Figure 8-1, a firewall's prevention function is complemented by the IDPS, which provides detection as well as prevention; automatic reactions by the IDPS and network administrators carry out the response.

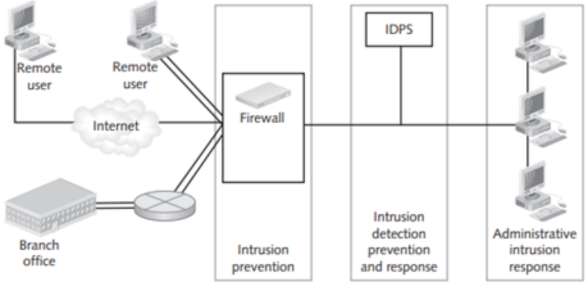

Remote user
Remote user
Internet
Firewall
IDPS
Branch office
Intrusion prevention
Intrusion detection prevention and response
Administrative intrusion response

Figure 8-1 The role of intrusion detection and prevention in network defense

It may seem that the goals of an IDPS are obvious: to alert administrators to breaches of a network or individual systems, and to prevent or minimize damage by reacting to the attack based on information collected by the IDPS. However, beyond the obvious goals are other considerations. It should be clear that IDPSs are only a small part of preventing intrusions. The use of firewalls, patching operating systems, and training users is also imperative to preventing network intrusions. Further, while it may not be possible to prevent an intrusion, the information gained by the IDPS might help prevent the same attack in the future.

An IDPS should be able to assess large volumes of network traffic or system activity to find signs of unauthorized access. An IDPS should also be able to record its findings in a log that allows administrators to examine past activity, and the system should be able to detect and record unauthorized access without compromise to produce evidence that will be admissible in court. Finally, an IDPS needs to be able to respond almost immediately to be effective.

An important goal of an IDPS is to make itself and the systems it protects as inaccessible as possible to attackers. With reasonable precautions, an IDPS can achieve relative security. Another value of deploying an IDPS is to demonstrate that an organization has made a good-faith effort to meet industry security standards and thus mitigate penalties associated with punitive damages if a harmful incident occurs.

COMMON DETECTION METHODOLOGIES

Anomaly and Signature Detection Systems

Currently, the three primary detection methodologies are signature detection, anomaly detection, and stateful protocol analysis. (Stateful protocol analysis, a relatively new approach, is discussed separately later in this section.) An anomaly detection system makes use of profiles that describe the services and resources each authorized user or group normally accesses on the network. Network baselines are also associated with profiles. Once these profiles are in place, the system can monitor users and groups for suspicious activity (anomalies) that does not fit the profiles. You might use anomaly detection if you are especially concerned about network misuse within the organization or you want to monitor all e-mail traffic, Web usage, and FTP servers.

Some IDPSs can create baselines during a “training period”—during this time, the IDPS monitors network traffic to observe what constitutes normal network behavior. Otherwise, you need to create profiles yourself. Because a large-scale network might consist of hundreds or thousands of users divided into many groups, profile configuration can require extensive work.

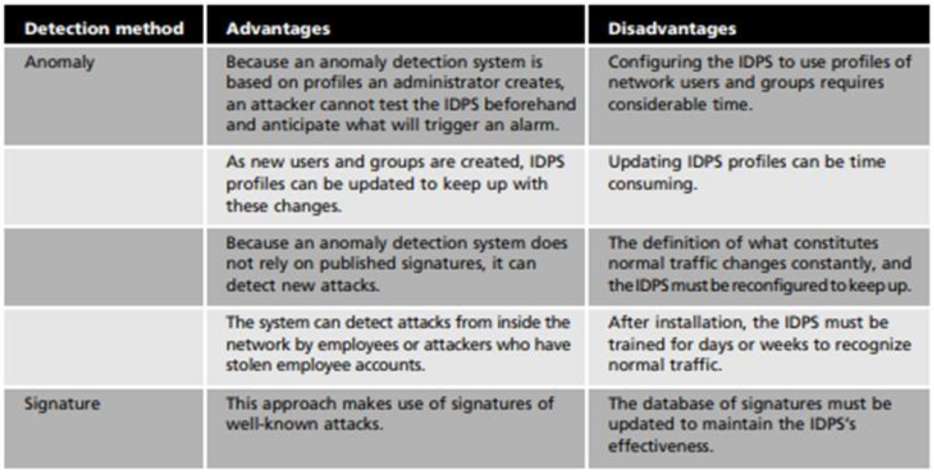
The accuracy of profiles has a direct impact on effective detection of anomalies. If profiles are accurate, the IDPS sends alarms only for genuine attacks. If profiles are incomplete or inaccurate, the IDPS sends alarms that turn out to be false positives—alarms generated by legitimate network traffic rather than actual attacks. False positives waste valuable time and resources; if they occur often enough, IT employees might not take alarms seriously. You also need to configure an IDPS accurately enough to avoid false negatives, genuine attacks that an IDPS does not detect. False negatives have the most serious security implications. True negatives are legitimate communications that do not set off an alarm. The term true positive is sometimes used to describe a genuine attack that an IDPS detects successfully.

An anomaly-based system can also generate false positives caused by changes in user habits; after all, people do not use computer systems in the same way all the time. When users vary a pattern (by attempting to access a database they have never used before, for instance), a false positive is likely.

In contrast to anomaly-based detection, which triggers alarms based on deviations from nor- mal network behavior, signature detection triggers alarms based on characteristic signatures of known external attacks. You might decide to use signature detection if you have the time and ability (and perhaps the software) to analyze the large amount of log file data this system generates.

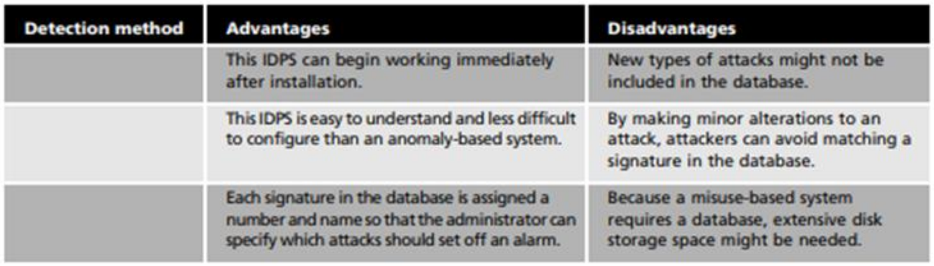
You might also see signature detection referred to as misuse-based detection or as a knowledge-based IDPS. The names for this detection technique vary, but the method is the same: Use databases of signatures of known attacks to identify and possibly respond to intrusions.

A signature-based IDPS is good for organizations that want a basic IDPS and are mostly concerned with known attacks from intruders trying to access internal hosts from the Internet. Network engineers research well-known attacks and record rules associated with each signature; a database of these signatures is then made available to an IDPS so that it can begin protecting networks immediately after installation. Signatures should be updated regularly. An anomaly-based IDPS, on the other hand, must be trained to recognize normal network traffic before it can protect a network. Table 8-1 summarizes the advantages and disadvantages of these detection systems.



Detection method	Advantages	Disadvantages
Anomaly	Because an anomaly detection system is based on profiles an administrator creates, an attacker cannot test the IDPS beforehand and anticipate what will trigger an alarm.	Configuring the IDPS to use profiles of network users and groups requires considerable time.
	As new users and groups are created, IDPS profiles can be updated to keep up with these changes.	Updating IDPS profiles can be time consuming.
	Because an anomaly detection system does not rely on published signatures, it can detect new attacks.	The definition of what constitutes normal traffic changes constantly, and the IDPS must be reconfigured to keep up.
	The system can detect attacks from inside the network by employees or attackers who have stolen employee accounts.	After installation, the IDPS must be trained for days or weeks to recognize normal traffic.
Signature	This approach makes use of signatures of well-known attacks.	The database of signatures must be updated to maintain the IDPS's effectiveness.

Table 8-1 Advantages and disadvantages of detection systems (continues)



Detection method	Advantages	Disadvantages
	This IDPS can begin working immediately after installation.	New types of attacks might not be included in the database.
	This IDPS is easy to understand and less difficult to configure than an anomaly-based system.	By making minor alterations to an attack, attackers can avoid matching a signature in the database.
	Each signature in the database is assigned a number and name so that the administrator can specify which attacks should set off an alarm.	Because a misuse-based system requires a database, extensive disk storage space might be needed.

Table 8-1 Advantages and disadvantages of detection systems (continued)

Monitoring an attack in progress can be helpful if you want to gather information for identifying, capturing, and prosecuting intruders or learn more about the vulnerability being exploited. The circumstances in which you allow an attack to progress should be spelled out clearly in the security policy. However, if the attack is causing severe harm, such as failure of network services or theft of proprietary data, stop the attack as quickly as possible.

Stateful Protocol Analysis

A signature-based system has another potential weakness you should keep in mind: the need to maintain state information (data about a connection) on a possible attack. This information gathering is called stateful protocol analysis. When an IDPS receives a packet, information about the connection between the host and remote computer is compared to entries in the state table. A state table maintains a record of connections between computers that includes the source IP address and port, destination IP address and port, and protocol. Furthermore, the IDPS needs to maintain state information for the entire length of the attack, which is called the event horizon. Maintaining this information might require an IDPS to review many packets of data; during long attacks, such as those that last from user logon to user logoff, the IDPS might not be able to maintain the state information long enough, and the attack could circumvent the system. Stateful protocol analysis can involve a variety of approaches:

- Traffic rate monitoring—If the IDPS detects a sudden and dramatic increase in traffic, such as that caused by a denial of service (DoS) attack, the IDPS can stop and reset all TCP traffic.

- Protocol state tracking (stateful packet filtering)—Some IDPSs can go a step beyond matching packet signatures by performing stateful packet filtering like that performed by firewalls. The IDPS maintains a record of the connection's state and allows packets to pass through to the internal network only if a connection has been established already.

- Dynamic Application layer protocol analysis—Sometimes an attacker can circumvent an IDPS by using a nonstandard port for an Application layer protocol. Dynamic Application layer protocol analysis involves detection of the protocol in use, followed by activation of analyzers that can identify applications not using standard ports.

- IP packet reassembly—Some IDPSs can reassemble fragmented IP packets to prevent fragments from passing through to the internal network.

Another detection method is heuristics—using an algorithm to detect suspicious traffic. Although this method is useful for detecting certain attacks, it is resource intensive and requires extensive tuning and maintenance to conform to the environment. Generally, heuristics are used for specific reasons and situations—for example, recognizing potential e-mail-based attacks. For most large networks, the drawbacks of heuristics often outweigh the advantages

Examining IDPS Components

Until relatively recently, IDSs (intrusion detection systems) did not contain prevention functions. The newer all-in-one devices, IDPSs, have become more effective and less costly and are now used widely. An IDPS has the advantage of increasing interoperability, addressing security concerns more thoroughly, and centralizing management. Some IDPS products can be quite complex, however, and they require extensive knowledge to configure and maintain.

The following sections describe typical components of an IDPS:

- Network sensors or host-based agents that analyze and report activity; they are used with management servers that receive and manage information from sensors, analyze data, and identify some events
- Detection and prevention capabilities
- A command console for interfacing with the IDPS
- A database server that stores attack signatures or behaviors an IDPS uses to identify potentially suspicious traffic

The National Institute of Standards and Technology (NIST) publishes a variety of resources, including NIST Special Publications, which are available at http://csrc.nist.gov/publications/nistpubs. This chapter is based on standards in NIST Special Publication 800-94, Guide to Intrusion Detection and Prevention Systems (IDPS).

Sensors and Agents

A sensor or agent functions as the electronic “eyes” of an IDPS. In host-based configurations, an IDPS installed on a single host computer has its agent built in to the IDPS software. In a network-based IDPS, a sensor is hardware or software that monitors network traffic in real time. (Host-based and network-based IDPS configurations are discussed later in the “Options for IDPSs” section.)

When a sensor detects an event it considers suspicious, an alarm is triggered. If appropriate, an automatic response may be initiated, such as blocking traffic from the source IP address of the attack. Attacks detected by an IDPS sensor can take one of two forms:

- Single-session attacks, in which an intruder makes an isolated attempt to locate a computer on the internal network or gain access by other means
- Multiple-session attacks, such as port scans or network scans, that take place over a period of time and are made up of several events

An IDPS that checks for network intrusions might have several sensors placed at strategic locations. Sensors should be placed at common entry points to the network (see Figure 8-2), such as the following:

- Internet gateways
- Connections between one network and another or between subnets separated by switches
- A remote access server that receives dial-up connections from remote users
- VPN devices that connect a LAN to a business partner’s LAN

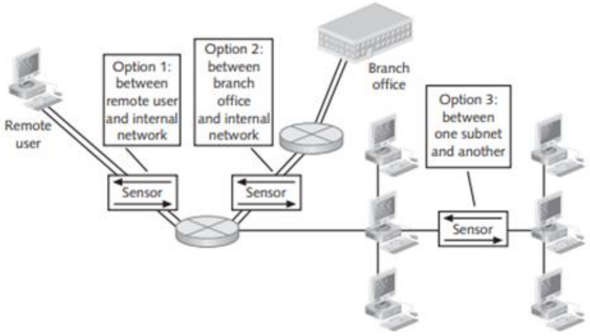

Remote user
Option 1: between remote user and internal network
Option 2: between branch office and internal network
Branch office
Option 3: between one subnet and another
Sensor
Sensor
Sensor

Figure 8-2 Positioning sensors at entry points to the network

In some IDPS configurations, sensor software collects data from a hardware device called a network tap, which gathers data from net- work traffic traveling over the physical media.

If a firewall is used to protect the network, sensors could be positioned on either side of the firewall. However, if the sensor is placed outside the firewall at a point exposed to the Inter- net, it could become the subject of an attack. A more secure location is behind the firewall in the demilitarized zone (DMZ), as shown in Figure 8-3.

Management Servers

An IDPS management server is the central repository for sensor and agent data. Sensors report log data to the management server, which analyzes and correlates the events received from several sensors. Some large networks have several management servers or even two tiers of servers, but small networks might not use any management servers.

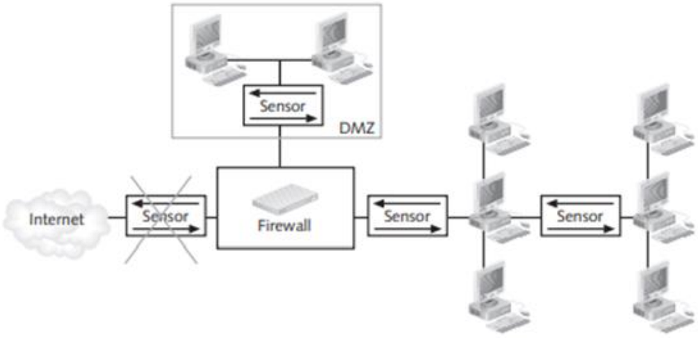


Figure 8-3 Positioning sensors behind the firewall in the DMZ

Detection and Prevention Capabilities

Most IDPSs support multiple detection capabilities for more flexibility in configuration and improved accuracy. When you are selecting an IDPS, consider the following customization options:

- Thresholds—Values that set the limit between normal and abnormal behavior
- Blacklists—Lists of entities, such as port numbers, URLs, or applications, that have been associated with malicious activity
- Whitelists—Lists of entities known to be harmless
- Alert settings—Specifying default priorities or severity levels, determining which prevention capabilities should be used for certain events, and specifying what information should be logged and how alert messages are sent, for example

Administrators can also view detection-related code, such as traffic signatures. With some IDPSs, administrators can review the application code of protocol-analysis programs. Some alerts might be generated from a complex set of signatures, so being able to modify IDPS signature code helps improve accuracy and is the only way to make the IDPS recognize characteristics that are specific to an organization. Writing custom IDPS signature code is complex and requires programming expertise because software bugs can cause an IDPS to malfunction or fail.

Any customization should be reviewed periodically to make sure it is still accurate and to account for changes in the environment; this advice also applies to baseline measurements used for anomaly detection. Keep in mind that IDPS updates and patches could affect custom settings.

Most modern IDPSs have multiple detection methods. Intrusion detection is still the basis of their operation, but these integrated systems often use compliance monitoring, too. An enterprise-class IDPS, such as McAfee Network User Behavior Analysis (www.mcafee. com/us/products/network-uba.aspx), monitors for vulnerabilities, access controls, logons, and host-level service behaviors, and it offers custom reporting and analysis tools. Most vendors also offer integrated management consoles for organizing data collected from several systems.

Prevention Capabilities

An IDPS can be configured to take preventive countermeasures, such as resetting all network connections when an intrusion is detected. Prevention capabilities vary by product. With some, administrators can specify what preventive measure should be taken for each alert type and decide whether to enable or disable prevention. Some IDPSs have a simulation mode in which all prevention capabilities are disabled, but the system generates reports that explain when the capabilities would be applied in response to different events. These reports are used to fine-tune prevention capabilities, which reduces the risk of blocking legitimate traffic.

IDPS prevention capabilities should not be considered a substitute for countermeasures taken by administrators, however. Administrators can use their judgment to determine whether an alarm is being triggered by a false positive or a genuine attack. If the attack is genuine, administrators can gauge its severity and determine whether the response should be escalated—increased to a higher level.

Command Console

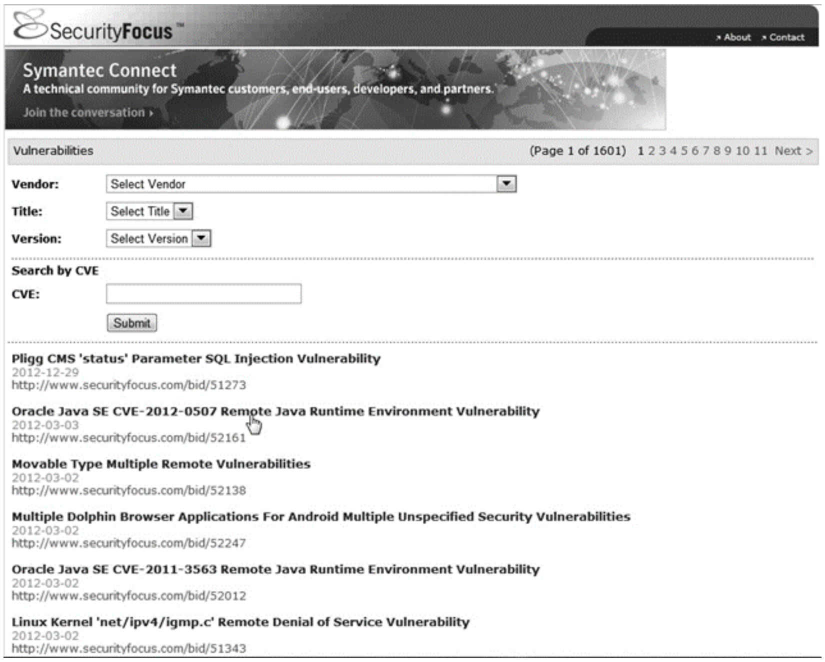
A command console is software that provides an interface to an IDPS. It enables administrators to receive and analyze alert messages and manage log files. In large-scale networks with more than one IDPS, a single console enables administrators to keep up with a large volume of events so that they can respond quickly. An enterprise-grade IDPS management product usually provides a single interface for analyzing and managing security events, and may also include multiple detection and prevention capabilities.

An IDPS can collect information from security devices throughout a management network, and they are connected to the command console where they can be evaluated. When a suspicious event is detected, the command console should not be slow to respond if its host computer is busy backing up files or performing firewall functions. Therefore, a command console is usually installed on a dedicated computer to maximize response speed.

A management network can be part of an organization's regular network, but ideally it should be separate. If this arrangement is not feasible, a good alternative is a virtual management network, which you can create by setting up a virtual LAN to segregate IDPS devices. This configuration provides some protection for the IDPS, but not as much as a full management network would provide.

Database of Attack Signatures or Behaviors

An IDPS does not have the ability to use judgment, so network administrators should exercise their own judgment when evaluating security alerts. However, an IDPS can use stored information to evaluate the traffic it monitors. Signature-detection IDPSs reference a database of known attack signatures; if a sensor detects a packet or a sequence of packets that match a signature, it sends an alert. The SecurityFocus online database of known vulnerabilities (see Figure 8-4) is updated frequently; you can search it to find information about a particular type of attack (http://online.securityfocus.com/bid).


SecurityFocus™
About
Contact
Symantec Connect
A technical community for Symantec customers, end-users, developers, and partners.
Join the conversation ›
Vulnerabilities
(Page 1 of 1601) 1 2 3 4 5 6 7 8 9 10 11 Next >
Submit

Vendor: Select Vendor

Title: Select Title

Version: Select Version

Search by CVE

CVE:

Pligg CMS 'status' Parameter SQL Injection Vulnerability
2012-12-29
http://www.securityfocus.com/bid/51273

Oracle Java SE CVE-2012-0507 Remote Java Runtime Environment Vulnerability
2012-03-03
http://www.securityfocus.com/bid/52161

Movable Type Multiple Remote Vulnerabilities
2012-03-02
http://www.securityfocus.com/bid/52138

Multiple Dolphin Browser Applications For Android Multiple Unspecified Security Vulnerabilities
2012-03-02
http://www.securityfocus.com/bid/52247

Oracle Java SE CVE-2011-3563 Remote Java Runtime Environment Vulnerability
2012-03-02
http://www.securityfocus.com/bid/52012

Linux Kernel 'net/ipv4/igmp.c' Remote Denial of Service Vulnerability
2012-03-02
http://www.securityfocus.com/bid/51343

Figure 8-4 The SecurityFocus online database of known vulnerabilities

The key to using attack signature databases is to keep them up to date; a new type of attack that has not been added to the database can defeat an IDPS quickly. An IDPS vendor that uses attack signatures should include a way to download new entries and add them to the database. The problem with systems that depend solely on signatures is that they are passive: They monitor traffic, compare it to the database, and send alerts whenever a packet matches a signature, which can result in many false positives. With most IDPSs, however, administrators can address this problem by adding custom rules to the signature database. An anomaly-based IDPS also uses a database of stored information for evaluating network traffic, so custom rules can be used with these systems to reduce false alarms.

Options for IDPSs

The preceding sections described different ways that an IDPS detects suspicious events and sends alarms. The following sections examine another way to categorize an IDPS: by its position on the network and how that position affects its activities. In the following sections, you learn about network-based IDPSs, host-based IDPSs, and hybrid IDPSs.

Network-Based IDPSs

A network-based IDPS (NIDPS) is a set of IDPS components that are specialized for network use. It examines traffic on network segments by using well-positioned sensors, management servers, a command console, and databases of signatures. It also has a mechanism for storing logs, backing up configurations, taking preventive action, and sending alert messages to designated administrators.

Sensors on an NIDPS can be hardware appliances or software. An appliance-based sensor usually has specialized network interface cards (NICs) for packet capture and processing, and for analyzing traffic signatures. NIDPS sensor appliances run an OS that is especially hardened and is not usually accessed by administrators. Software-based sensors can be installed on hardened hosts or contain a specialized OS designed to run on hosts with specific configurations.

Positioning an NIDPS on the Network

NIDPS sensors are commonly installed behind the firewall and before the LAN, between the firewall and the DMZ, or on any network segment (see Figure 8-5).

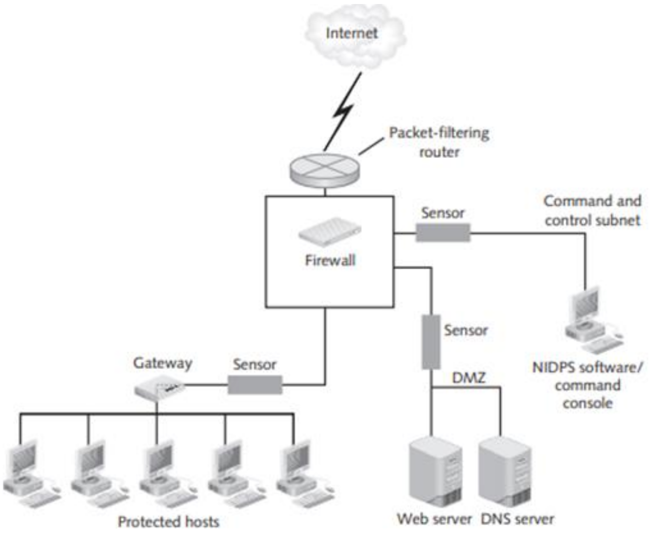


Figure 8-5 An NIDPS monitoring traffic behind the firewall or in the DMZ

Positioning sensors at the network perimeter is ideal for enabling the IDPS to sniff packets (receive and analyze them) as they pass into the network. Each IDPS sensor is also equipped with its own NIC so that it can sniff packets in promiscuous mode, in which each packet is detected and analyzed in its entirety.

An NIDPS can use inline sensors or passive sensors. An inline sensor is positioned so that network traffic must pass through it. This type of sensor is used to stop attacks from blocking network traffic and is usually placed where firewalls and other security devices are positioned, such as between network segments or at connections to external networks. The drawback of inline sensors is the potential to create a traffic bottleneck if the sensor becomes overloaded, but you have two possible workarounds for this problem. You can position the inline sensor on the more secure side of the network, such as behind the firewall, so that it has less traffic to process. You can also place the sensor on the less secure side to protect and lessen the load on the device that divides the net- works, such as a router. Figure 8-6 shows an inline sensor positioned inside the firewall perimeter.

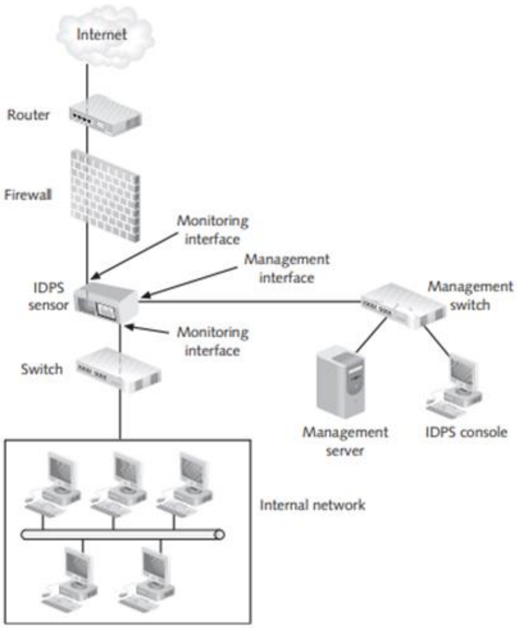


Figure 8-6 Positioning an inline sensor

Passive sensors monitor copies of actual traffic; no actual traffic passes through them. They are typically placed at key network locations, such as divisions between networks, or on key network segments, such as the DMZ (see Figure 8-7). Passive sensors monitor traffic by using the following methods:

- Spanning port—A port on many switches that can see all network traffic passing through. A passive IDPS sensor attached to this port can monitor traffic flow without the traffic actually passing through it.
- Network tap—A direct connection between a sensor and the physical network medium, such as a fiber-optic cable.
- IDPS load balancer—A device that collects and directs traffic to monitoring systems. Administrators configure rules that tell the load balancer where to direct different types of traffic.

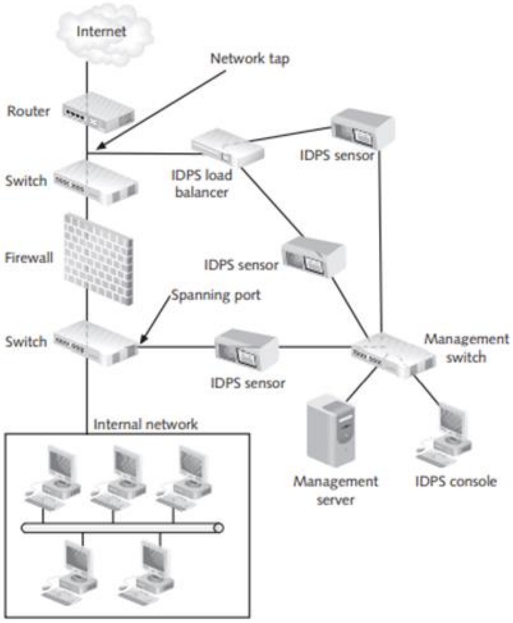


Figure 8-7 Positioning a passive sensor

NIDPS Capabilities

NIDPS capabilities vary depending on the product. Some can collect information about hosts, OSs, applications, and network activities and characteristics. By identifying hosts, you could develop a list of all hosts on the network, organized by IP or MAC addresses. Tracking which ports are used on each host can indicate which OS is running, and knowing which OS versions are used can also help identify vulnerable hosts. An NIDPS can also analyze packet headers to identify characteristics or unusual behavior of OSs. By identifying an application, you can monitor which ports are being used and help identify misuse of the application. Collecting general information about network characteristics, such as hop count, can help identify changes to the network configuration, such as when a new device is added or removed.

A primary function of an NIDPS is extensive logging of traffic. You can use traffic logs to identify and analyze potential attacks, locate vulnerabilities, assess network use and performance, and correlate with other device logs. NIDPSs usually log the following types of information:

- Timestamps
- Event or alert types
- Protocols
- Connection or session IDs
- Source and destination IP addresses and ports
- Size of transmissions, usually in bytes
- State-related information
- Application requests and responses
- Network, Transport, and Application layer protocols
- Preventive action taken, if any

Most NIDPSs use a variety of detection capabilities, which are usually combinations of anomaly detection, signature detection, and stateful protocol analysis. These capabilities work together to improve detection efficiency and accuracy. For example, stateful inspection might parse activity into requests and responses, which are then examined for anomalies and compared to signatures. This type of multilayered inspection is similar to the defense- in-depth strategy, and is better for accuracy than using a single detection method.

The prevention capabilities of an NIDPS vary depending on the product and the sensor type used. An NIDPS has one of the following sensor types:

- Passive only—Ends the current TCP session (called session sniping, not commonly used now)
- Inline only—Uses inline firewalling and bandwidth throttling, and alters malicious content
- Passive and inline—Reconfigures other network security devices, perhaps by instructing a firewall to block certain types of activity or running an administrator-defined script

With most NIDPSs, administrators can configure specific actions for each type of alert. Some also offer simulation modes for fine-tuning alerts and prevention actions.

NIDPS Management

After choosing an NIDPS, an administrator should design the architecture and then test and secure the NIDPS components. Designing the architecture includes determining where sensors are located, how many are needed, and how they should be connected. Testing NIDPS components includes accounting for network downtime while deploying sensors or network taps or activating spanning ports. Securing NIDPS components involves making sure that sensors are not assigned IP addresses on monitoring inter- faces so that other hosts cannot initiate connections to them. Securing components also requires hardening management networks and configuring hosts for log files and backups.

As with any networking component, NIDPS architecture, detection, and prevention configurations should be evaluated regularly. NIDPS software and hardware systems must be updated and patched periodically, and custom configurations should be monitored and adjusted as needed to remain effective against network threats

Host-Based IDPSs

In contrast to an NIDPS on the network perimeter, a host-based IDPS (HIDPS) is deployed on hosts inside the network perimeter. On a small network, HIDPS features might be hosted on the machine that an HIDPS is monitoring. The HIDPS host could be a printer, Web server, computer, firewall, switch, router, or combination. HIDPSs are generally deployed only on sensitive or mission-critical hosts because placing one on every host in the network would be very expensive. HIDPSs in a large network environment commonly use management servers, signature databases, and consoles for configuration and monitoring. Often, an HIDPS uses the same IDPS infrastructure as an NIDPS. Figure 8-8 shows a typical HIDPS deployment.

An HIDPS can also consist of dedicated appliances running agent software that are positioned to monitor traffic on a particular host. Technically, this arrangement could be considered an inline NIDPS, but it is categorized as an HIDPS because it is highly specialized for monitoring a single host or type of traffic. An HIDPS appliance is often used to protect a Web server or database server.

An HIDPS monitors and evaluates packets generated by the host and gathers data from OS and application logs on the host. An HIDPS also gathers system variables such as the following:

- System processes
- CPU use
- File accesses
- System logs
- System and application configuration changes

System events that match the signatures of known attacks reach the IDPS on the host, which sends an alert message to users or the administrator. An HIDPS does not sniff packets as they enter the network. Instead, it monitors log file entries and user activity, and is highly effective at tracking misuse of resources by internal users.

Configuring an HIDPS

An HIDPS can have two configurations: centralized or distributed. In a centralized configuration, the HIDPS sends all data it has gathered to a central location such as the command console or management server for analysis. In a distributed configuration, data analysis is distributed among hosts; each host analyzes data and sends only alert messages (not the data) to the command console.

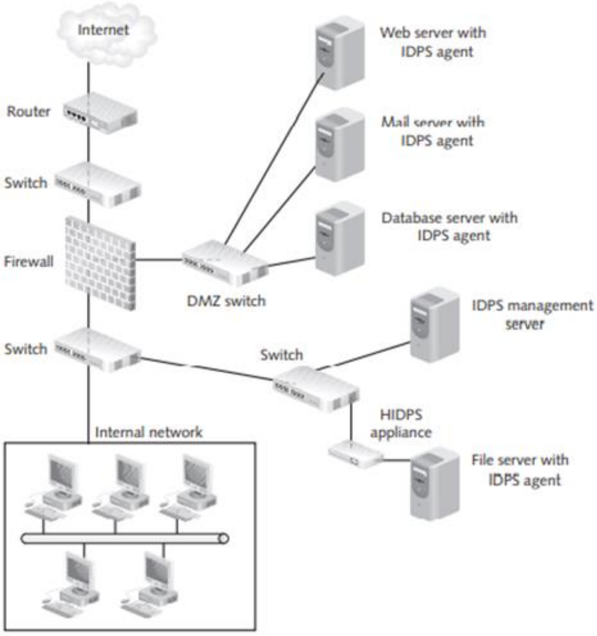

Internet
Router
Switch
Firewall
DMZ switch
Switch
Internal network
Web server with IDPS agent
Mail server with IDPS agent
Database server with IDPS agent
IDPS management server
Switch
HIDPS appliance
File server with IDPS agent

Figure 8-8 A typical HIDPS deployment

In a centralized configuration (see Figure 8-9), the host's performance is unaffected by the IDPS. However, because data is sent to the command console for analysis, alert messages do not occur in real time.

The process in Figure 8-9 is as follows:

1. An event is generated on the host.
2. Data gathered by the IDPS agent (a software program running on the host) is transmitted to the command console for analysis.
3. A log file entry is created.
4. If necessary, an alert is generated.
5. The IDPS responds.
6. Finally, data is stored in a database for long-term analysis.

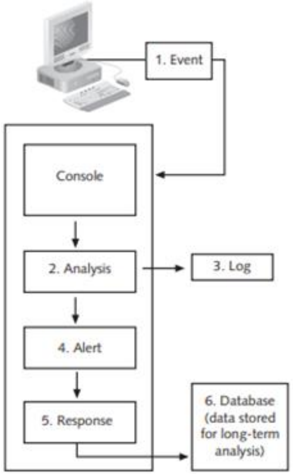


Figure 8-9 A centralized HIDPS

In a distributed configuration, event data processing is distributed between the host and command console. The host generates the data and analyzes it in real time. As a result, analysis can be performed without a delay, but the trade-off is reduced performance on the host. The host processes all data whether alerts are required or not. Data is then transmitted to   the command console in the form of alert messages, as shown in Figure 8-10.

Choosing the Host

The RAM, hard disk space, and processor speed required on the host depend on the type of HIDPS you use. In a centralized configuration, processing is per- formed on the command console, so the host's performance requirements are minimal. However, in a distributed configuration, the host gathers intrusion data and analyzes it in real time, so it needs the maximum memory and processor speed. Check the vendor's IDPS requirements for specific details.

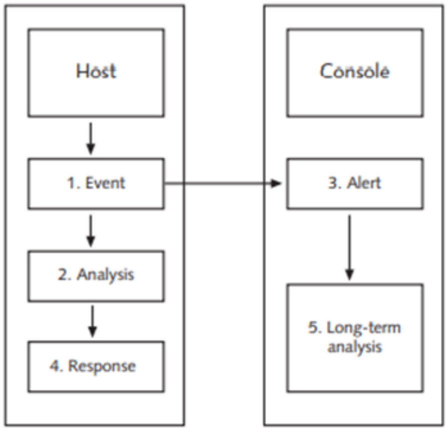


Figure 8-10 Processing event data from an HIDPS

Comparing an NIDPS and HIDPS

An HIDPS can tell you whether an attack attempt on the host was successful. An NIDPS, in contrast, provides alerts on suspicious network activity but does not tell you whether an attack attempt reached the targeted host and whether an intrusion actually occurred. An HIDPS can detect attacks that would get past an NIDPS. For example, fragmentation, out-of-sequence techniques, or other masking techniques might bypass firewalls and NIDPSs as legitimate traffic.

On the other hand, an HIDPS provides only data pertaining to the host on which it is installed, not the network as a whole. An HIDPS cannot detect an intrusion attempt that targets the entire network, such as a port scan on a range of computers. If you use an HIDPS, you need to install it on several hosts on the network, which takes time and can be more expensive than an NIDPS.

In addition, an HIDPS can compare records stored in audit logs to detect inconsistencies in how applications and systems programs are used. However, they are susceptible to some DoS attacks and could create increased performance overhead on host systems.

Hybrid IDPSs

A hybrid IDPS combines the capabilities of an HIDPS and NIDPS for more flexibility and security. The challenge in using a hybrid IDPS is getting the components to work together, although many IDPS products have built-in hybrid capabilities. Variations of a hybrid IDPS—combined IDPS sensor locations and combined IDPS detection methods—are described in the following sections.

Combining IDPS Sensor Locations

One type of IDPS hybrid combines host-based and network-based systems. The combination enables sensors to be positioned on network segments and as agents on hosts. As a result, the network can report on attacks aimed at network segments or the network as a whole. In addition, computers that store confidential information, such as databases of job records, can be protected with an HIDPS. An IDPS on a host, especially one with a distributed configuration, can analyze data in real time and send an alert immediately that notifies the administrator of a possible unauthorized access attempt.

Combining IDPS Detection Methods Another type of IDPS hybrid combines anomaly and signature detection. The combination helps overcome the limitations of each detection method in the following ways:

· Having a database of known attack signatures enables the system to start running immediately and effectively wards off most well-known external attack methods.

· Having an anomaly-based system keeps the system flexible and capable of detecting internal misuse that deviates from normal use patterns.

A hybrid IDPS that combines anomaly and signature detection can respond to the latest, previously unreported attacks and to attacks from both external and internal sources. A drawback is that administrators must do more configuration and coordination work. Data from multiple sources must be collected in a central location where it can be reviewed and analyzed.

Advantages and Disadvantages of a Hybrid IDPS

A hybrid IDPS has the advantage of combining aspects of NIDPS and HIDPS configurations. You can monitor the network as a whole with network-based sensors and monitor attacks that reach computers with host-based sensors. The drawback of a hybrid IDPS is the problem of getting disparate systems to work in a coordinated fashion. In addition, data gathered from multiple systems can be difficult to analyze easily.

Securing IDPS Components

As you might expect, IDPSs are a concern for attackers; efforts to circumvent detection are a prime focus of professional hackers. While thrill-seeking or hacktivist attackers may want to have their presence known to increase their status with other h ackers or to gain attention for their cause, sophisticated criminal attackers want to remain hidden. A compromised IDPS gives hackers much more flexibility in crafting their attacks.

In some cases, a stealthy attack may not be possible, and the attacker may disable an IDPS with a denial of service attack before moving on to targets that are more valuable. One approach to address this type of attack is to create a hierarchical IDPS architecture based on mobile agent technologies so that the IDPS components can relocate from compromised hosts to intact hosts.

IDPS Security Best Practices

· An IDPS must be able to handle the volume of traffic or activity it encounters so that it does not drop packets. Providing adequate throughput helps secure the IDPS and the network assets.

· IDPSs should be tested regularly. Testing should include security capabilities, logging, performance, and management system operation.

· Sensors should not be addressable. In other words, although sensors can capture packets, they should not have an IP address on the network they are monitoring. The addressable interface should be only on the management subnet.

· Communication between IDPS components should be encrypted.

- Authentication should be required for use and administration of the IDPS, and access control and auditing should be implemented.
- IDPSs should be able to continue operating during denial of service attacks.
- Remote logging should be used in an HIDPS.
- Operating systems of HIDPSs should be patched and hardened.

Developing IDPS Filter Rules

You have learned that an IDPS can use specific signatures of malicious packets to detect attacks. As you might imagine, these rules can get complicated. While rules to detect network scans may be relatively simple, rules to detect the preparatory steps of an attack through e-mail or the Web are much more involved. In the hands-on projects at the end of this chapter, you will download a free IDPS called Snort, and then create your own rule and test it.

Before you create your own rule, however, you need to learn the basics of Snort rule syntax. Each Snort rule has two sections: the rule header and the rule options. To examine these sections, use the following example:

```
alert tcp  any any -> 192.168.21.0/24  111  (content:“00 01  86 a5”; msg:“mountd access”;)
```

The header is the opening portion in the example; the options are within the parentheses. The first field in the header is the action the system takes when a packet is detected that meets the requirements of the rule. In this case, the action field is “alert,” meaning that an alert will be logged. More sophisticated alert actions are available; for example, a message could be sent to an administrator’s desktop or e-mail account. Another action might be “log,” meaning that no alert would be registered or sent, but the contents of the packet would be recorded in the log file. The next field defines the protocol; this example specifies the TCP protocol, but other protocols can be used, like UDP or ICMP.

The next two fields define the source system’s socket—the source IP address and the source port, respectively. In the example, “any” is specified for both fields, meaning that a packet sent from any IP address and any port address would fit the rule specifications. This approach makes sense if you want to identify a type of packet regardless of whether it originates inside or outside your network. The next field is an arrow that indicates the source and destination systems; the arrow always points from the source to the destination system.

Next is the destination IP address field. In the preceding example, the specification “192.168.21.0/24” is a network address. Thus, if any host on the 192.168.21.0 network received a TCP packet from any system, this rule would apply. When specifying IP addresses, CIDR (Classless Interdomain Routing) notation is required. In CIDR notation, the IP address is followed by a forward slash and then the number of network identifier bits in the subnet mask. In the preceding example, the /24 means that the first 24 bits constitute the mask. The dotted decimal notation for /24 is 255.255.255.0. You are not limited to using a single arrow between IP addresses; you can also use the “<>” symbols to indicate that the rule applies to traffic moving in either direction.

The destination port field follows; port 111 is specified in the preceding example. This specifi- cation makes the rule much more precise because it instructs Snort to ignore any packets that are not being sent to port 111, even if all the other rule parameters are matched. This field completes the rule header section.

The options portion of the rule specifies detailed characteristics of the frame and more specific Snort actions. In the example, two options are used: “content” and “msg.” The content option means that any packet containing specified content—in this case, the hexadecimal string 00 01 86 a5—will trigger the rule, assuming that the other specifications match. The second option, msg, instructs Snort to add the text that follows to the captured packet in the log. Here, any packet that meets the previous specifications will have the words mountd access added to the top of the packet shown in the log file, making it easier for administrators or software-auditing programs to spot the message when reviewing logs. All options must be listed within parentheses. The option name must be followed by a colon and the option must end with a semicolon, even if it is the last option listed. Snort is very particular; if your rule contains a syntax error, it will fail.

You can use an exclamation point to negate a statement. In the following example, the “!” means not to log any traffic from or to any host on the internal network (192.168.21.0/24):

```
log !192.168.21.0/24 any <> 192.168.21.0/24 any
```

Now that you have sufficient background, you can write your own rules in the hands-on pro- jects later in this chapter.

Examining Intrusion Detection Step by Step

IDPSs operate in different ways depending on whether they are configured to react to anomalies or signatures. Despite these differences in operation, the process can be divided into general steps, shown in Figure 8-11, that apply to most IDPSs. These steps are described in the following sections.

Step 1: Installing the IDPS Database

The first step of intrusion detection occurs before any packets are detected on the network. Along with the IDPS software and hardware, you need to install the database of signatures or user profiles, which gives the IDPS a set of criteria against which it can compare packets passing through the sensor.

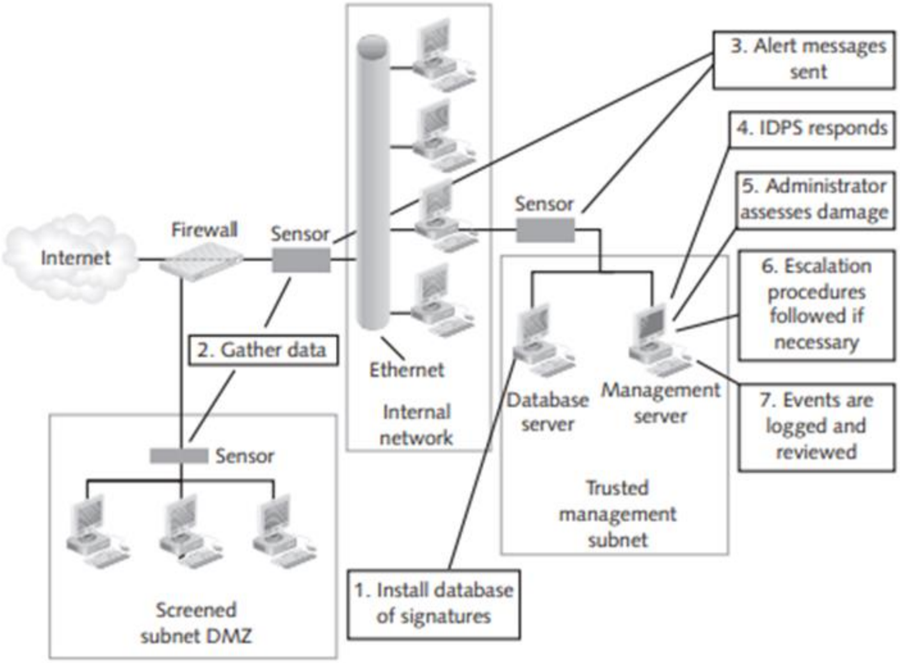

3. Alert messages sent
4. IDPS responds
5. Administrator assesses damage
6. Escalation procedures followed if necessary
7. Events are logged and reviewed
Internet
Firewall
Sensor
Sensor
Sensor
2. Gather data
Ethernet
Internal network
Database server
Management server
Trusted management subnet
1. Install database of signatures
Screened subnet DMZ

Figure 8-11 Steps in intrusion detection

In an anomaly-based system, installing the database can take up to a week longer than installing other IDPS devices so that the IDPS can observe network traffic and compile baseline data on normal network use. Some data can take a week to record because it occurs over a period of days—for example, a series of daily logons to the network. In a signature-based IDPS, you can install the database of attack signatures included with the software, or you can add your own custom rule base to account for new attacks or special situations that generate false positives.

Step 2: Gathering Data

After the IDPS and database are installed, network sensors and agents can gather data by reading packets. Agents installed on hosts observe traffic entering and leaving the hosts; sensors placed on network segments read packets that pass through those segments.

Sensors need to be positioned where they can capture all packets entering and leaving a host or network segment. Sensors placed on network segments cannot always capture every packet if the traffic level becomes too heavy, however. Repositioning agents on each network host improves accuracy, even though the expense of purchasing new agents and the effort of installing them can be considerable. The most important consideration is being able to capture all packets so that none can circumvent the IDPS.

Step 3: Sending Alert Messages

The IDPS detection software compares packets it observes with signatures stored in its data- base. An alert message is transmitted when a packet matches an attack signature or deviates from normal network use. The alert message goes to the IDPS command console, where the network administrator can evaluate it.

Step 4: The IDPS Responds

When the command console receives an alert message, it notifies the administrator using a method that the administrator has configured. The console might display a pop-up window or send an e-mail, for instance. An IDPS can also be configured to take action when a suspicious packet is received and an alert message is sent. The following list describes typical preventive or response actions an IDPS can take:

- Alarm—An alert message is sent to the command console or another designated location.
- Drop—The packet is dropped without an error message being sent to the originating computer.
- Reset—The IDPS is instructed to stop and restart network traffic, thus halting severe attacks.
- Code analysis—The IDPS can prevent malicious code from running or stop certain applications from opening shells used to launch some attacks.
- File system monitoring—The IDPS can prevent files from being modified, accessed, replaced, or deleted.
- Network traffic filtering—The IDPS can act as a host-based firewall to stop violations of unauthorized access or acceptable use policies, based on IP address, protocol, or port information.
- Network traffic analysis—The IDPS can stop incoming traffic from reaching a host or leaving it. This action is useful for stopping Network layer, Transport layer, or Application layer attacks as well as unauthorized applications and protocols.

An IDPS stops TCP traffic by sending a TCP packet with the RST (reset) flag set, which terminates the connection with the computer that is attempting to attack the system. Resetting TCP traffic does not affect UDP or other types of traffic, however. This action, called session sniping, is not used often in a modern IDPS.

Step 5: The Administrator Assesses Damage

An automated response sent by an IDPS is like a call to action. The administrator has the responsibility to monitor alerts and determine whether countermeasures need to be taken. When an IDPS is first installed, it might send many alerts that are false positives, depending on the accuracy of information in the database. Administrators usually have to fine-tune databases to account for situations that seem to be intrusions but are actually legitimate traffic.

In an anomaly-based system, for example, an adjustment might be needed for an employee who logs on over the weekend instead of during the standard workweek. In a misuse-based system, an adjustment can be made to allow traffic that the IDPS might determine to be suspicious, such as a vulnerability scan performed by a device at a particular IP address. You could add a rule that changes the IDPS action in response to traffic from that IP address. Figure 8-12 shows the dividing line between acceptable and unacceptable network use.

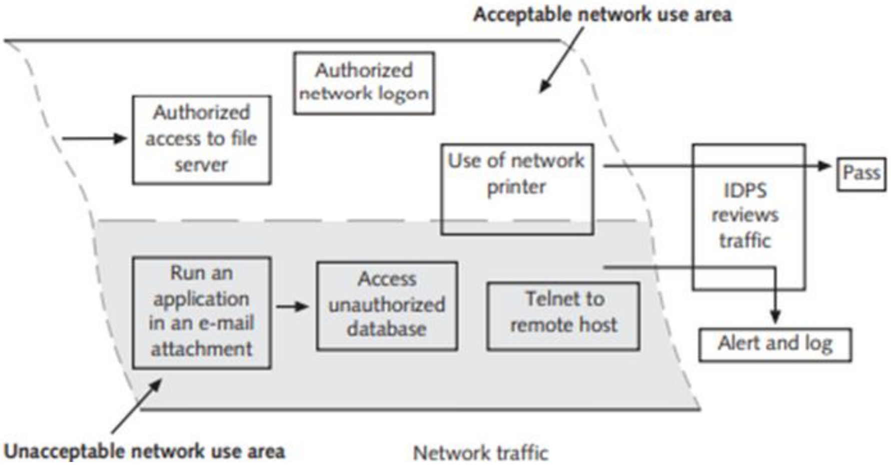

Acceptable network use area
Authorized network logon
Authorized access to file server
Use of network printer
IDPS reviews traffic
Pass
Run an application in an e-mail attachment
Access unauthorized database
Telnet to remote host
Alert and log
Unacceptable network use area
Network traffic

Figure 8-12 Differentiating acceptable and unacceptable network use

The line dividing acceptable network use from unacceptable use is not always clear, however. In Figure 8-12, for example, the box that indicates network printer use overlaps into the unacceptable area because acceptable use of the printer is limited to specific purposes. Printing office-related documents constitutes acceptable use, but printing personal photos probably falls on the unacceptable side of the line.

The goal of adjusting the IDPS database is not to avoid false positives because they are almost inevitable. False positives do consume an administrator’s time and energy, but they do not compromise network security. The goal is avoiding false negatives—incidents that should cause an alarm but do not. False negatives occur without anyone’s knowledge and are a potentially serious breach of security. Although false positives are often seen as nuisances, they are far less serious than false negatives.

Step 6: Following Escalation Procedures

Escalation procedures are a set of actions that are spelled out in the security policy and followed if the IDPS detects a true positive (a legitimate attack). These procedures vary depending on the severity of the incident. A Level One incident might be managed quickly with only a single security professional. A Level Two incident represents a more serious threat and must be escalated to involve a security professional with more authority. A Level Three incident represents the highest degree of threat.

Step 7: Logging and Reviewing Events

After an IDPS has sent an alert to the command console and responded as necessary, the event that caused the alert is entered in the IDPS log. The event can also be sent to a data- base file, where it can be reviewed with other previous alerts. Reviewing a number of alerts sent over a period of time enables administrators to determine whether patterns of misuse have occurred. This review also gives administrators the opportunity to spot a gradual attack, such as a series of logons occurring only once every few days or a series of ping sweeps occurring once a week over a few months.

An IDPS should also provide accountability—the ability to track an attempted attack or intrusion back to the responsible party. Some systems have a built-in tracing feature that attempts to locate the IP address associated with an event. Identifying the person who used the source computer can be difficult, but a trace can at least provide a starting point for identifying an attacker.

Evaluating IDPS Products

Some IDPS products include antivirus or firewall capabilities, and IDPS products can be integrated and combined for comprehensive protection. The options for custom defenses that include an IDPS are numerous, so you should focus on evaluating the system that best fits your needs.

As with any network security component, the most expensive or full-featured product is not always the best option. Important considerations include the availability of staff to install and support technologies, the organization’s security stance, and the operating environment. The best approach is to know what you need before you invest resources. You should consider the following basic factors when evaluating IDPS products:

- Determine whether an IDPS is necessary. Sometimes, existing network components can perform IDPS functions well enough to meet security requirements.
- Conduct a risk assessment to decide which network resources require protection and at what level.
- Define general requirements and goals that the IDPS should meet. Assess security poli- cies and other related organizational policies that affect security and IDPS goals.
- Determine whether it is acceptable to use proprietary products or open-source products.
- Consider the frequency and accuracy of signature updates.
- Assess the availability of support.
- Evaluate the technical specifications of IT systems so that you can estimate how many products are needed and where they should be positioned.
- Determine your external security requirements, such as applicable laws, security audit requirements, results of investigated security incidents, or organizational cryptography specifications.
- Evaluate your needs for security capabilities, information gathering, and logging.
- Review the detection and prevention capabilities of the IDPS products you are considering.
- Identify performance and management requirements for the IDPS.
- Define the interoperability and scalability potential for each product you are considering.
- Determine a reasonable cost estimate that includes acquisition, testing, installation, daily use, maintenance, and updates to keep the IDPS operating at peak efficiency.
- Identify resource limitations, such as budget, staff, or hardware.
- Identify any training, documentation, and support required for IDPS products.