1. Intrusion Detection and Prevention Systems

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. IDS and IPS technologies offer many of the same capabilities, and administrators can usually disable prevention features in IPS products, causing them to function as IDSs. Accordingly, for brevity the term intrusion detection and prevention systems (IDPS) is used throughout the rest of this chapter to refer to both IDS and IPS technologies. Any exceptions are specifically noted.

This chapter provides an overview of IDPS technologies. It explains the key functions that IDPS technologies perform and the detection methodologies that they use. Next, it highlights the most important characteristics of each of the major classes of IDPS technologies. The chapter also discusses IDPS interoperability and complementary technologies.

1.1 Fundamental Concepts

IDPSs are primarily focused on identifying possible incidents. For example, an IDPS could detect when an attacker has successfully compromised a system by exploiting a vulnerability in the system. The IDPS would log information on the activity and report the incident to security administrators so that they could initiate incident response actions to minimize damage. Many IDPSs can also be configured to recognize violations of acceptable use policies and other security policies—examples include the use of prohibited peer-to-peer file sharing applications and transfers of large database files onto removable media or mobile devices. Additionally, many IDPSs can identify reconnaissance activity, which may indicate that an attack is imminent or that a certain system or system characteristic is of particular interest to attackers. Another use of IDPSs is to gain a better understanding of the threats that they detect, particularly the frequency and characteristics of attacks, so that appropriate security measures can be identified. Some IDPSs are also able to change their security profile when a new threat is detected. For example, an IDPS might collect more detailed information for a particular session after malicious activity is detected within that session.

IPS technologies differ 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 include the IPS terminating the network connection being used for the attack and the IPS blocking access to the target from the offending user account, IP address, or other attacker attribute.
- The IPS changes the security environment. The IPS could change the configuration of other security controls to disrupt an attack. Common examples are the IPS reconfiguring a network firewall to block access from the attacker or to the target, and the IPS 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 email and then permitting the cleaned email 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.

Some IPS sensors have a learning or simulation mode that suppresses all prevention actions and instead indicates when a prevention action would have been performed. This allows administrators to monitor and fine-tune the configuration of the prevention capabilities before enabling prevention actions, which reduces the risk of inadvertently blocking benign activity.

A common attribute of all IDPS technologies is that they cannot provide completely accurate detection. Incorrectly identifying benign activity as malicious is known as a false positive; the opposite case, failing to identify malicious activity, is a false negative. It is not possible to eliminate all false positives and negatives; in most cases, reducing the occurrences of one increases the occurrences of the other. Many organizations choose to decrease false negatives at the cost of increasing false positives, which means that more malicious events are detected but more analysis resources are needed to differentiate false positives from true malicious events. Altering the configuration of an IDPS to improve its detection accuracy is known as tuning.

Most IDPS technologies also compensate for the use of common evasion techniques. Evasion is modifying the format or timing of malicious activity so that its appearance changes but its effect is the same. Attackers use evasion techniques to try to prevent IDPS technologies from detecting attacks. For example, an attacker could encode text characters in a particular way that the target will understand, hoping that IDPSs monitoring the activity will not. Most IDPS technologies can overcome evasion techniques by duplicating special processing performed by the targets.

1.1.1 IDPS Detection Methodologies

IDPS technologies use many methodologies to detect attacks. The primary classes of detection methodologies are signature-based, anomaly-based, and stateful protocol analysis, respectively. Most IDPS technologies use multiple methodologies, either separately or integrated, to provide more broad and accurate detection. These methodologies are described in more detail below.

Signature-Based Detection

A signature is a pattern that corresponds to a known attack or type of attack. Signature-based detection is the process of comparing signatures against observed events to identify possible attacks. Examples of signatures are:

- A telnet attempt with a username of "root", which is a violation of an organization's security policy

- An email with a subject of “Free pictures!” and an attachment filename of “freepics.exe”, which are characteristics of a known form of malware
- An operating system log entry with a status code value of 645, which indicates that the host’s auditing has been disabled.

Signature-based detection is very effective at detecting known attacks but largely ineffective at detecting previously unknown attacks, attacks disguised by the use of evasion techniques, and many variants of known attacks. For example, if an attacker modified the malware in the previous example to use a filename of “freepics2.exe”, a signature looking for “freepics.exe” would not match it.

Signature-based detection is the simplest detection method because it just compares the current unit of activity, such as a packet or a log entry, to a list of signatures using string comparison operations. Detection technologies that are solely signature-based have little understanding of many network or application protocols and cannot track and understand the state of communications—for example, they cannot pair a request with the corresponding response, nor can they remember previous requests when processing the current request. This prevents signature-based methods from detecting attacks that comprise multiple events if no single event contains a clear indication of an attack.

Anomaly-Based Detection

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDPS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. The profiles are developed by monitoring the characteristics of typical activity over a period of time. For example, a profile for a network might show that Web activity comprises an average of 13% of network bandwidth at the Internet border during typical workday hours. The IDPS then uses statistical methods to compare the characteristics of current activity to thresholds related to the profile, such as detecting when Web activity comprises significantly more bandwidth than expected and alerting an administrator of the anomaly. Profiles can be developed for many behavioral attributes, such as the number of emails sent by a user, the number of failed login attempts for a host, and the level of processor usage for a host in a given period of time.

The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown attacks. For example, suppose that a computer becomes infected with a new type of malware. The malware could consume the computer’s processing resources, send many emails, initiate large numbers of network connections, and perform other behavior that would be significantly different from the established profiles for the computer.

An initial profile is generated over a period of time sometimes called a training period. Profiles can either be static or dynamic. Once generated, a static profile is unchanged unless the IDPS is specifically directed to generate a new profile. A dynamic profile is adjusted constantly as additional events are observed. Because systems and networks change over time, the corresponding measures of normal behavior also change; a static profile will eventually become inaccurate, so it needs to be regenerated periodically. Dynamic profiles do not have this problem, but they are susceptible to evasion attempts from attackers. For example, an attacker can perform small amounts of malicious activity occasionally, then gradually increase the frequency and quantity of activity. If the rate of change is sufficiently slow, the IDPS might think the malicious activity is normal behavior and include it in its profile.

Another problem with building profiles is that it can be very challenging in some cases to make them accurate because computing activity is so complex. For example, if a particular maintenance activity that performs large file transfers occurs only once a month, it might not be observed during the training period; when the maintenance occurs, it is likely to be considered a significant deviation from the profile. Anomaly-based IDPS products often produce many false positives because of benign activity that deviates significantly from profiles, especially in more diverse or dynamic environments. Another noteworthy problem with the use of anomaly-based detection techniques is that it is often difficult for analysts to determine what triggered a particular alert.

Stateful Protocol Analysis

Stateful protocol analysis is the process of comparing predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations. Unlike anomaly-based detection, which uses host or network-specific profiles, stateful protocol analysis relies on vendor-developed universal profiles that specify how particular protocols should and should not be used. The “stateful” in stateful protocol analysis means that the IDPS is capable of understanding and tracking the state of network, transport, and application protocols that have a notion of state. For example, when a user starts a File Transfer Protocol (FTP) session, the session is initially in the unauthenticated state. Unauthenticated users should only perform a few commands in this state, such as viewing help information or providing usernames and passwords. An important part of understanding state is pairing requests with responses, so when an FTP authentication attempt occurs, the IDPS can determine if it was successful by checking the status code in the corresponding response. Once the user has authenticated successfully, the session is in the authenticated state, and users are expected to perform any of several dozen commands. Performing most of these commands while in the unauthenticated state would be considered suspicious, but in the authenticated state performing most of them is considered benign.

Stateful protocol analysis can identify unexpected sequences of commands, such as issuing the same command repeatedly or issuing a command without first issuing another command upon which it is dependent. Another state tracking feature of stateful protocol analysis is that the IDPS can keep track of the authenticator used for each session, and record the authenticator used for suspicious activity. Some IDPSs can also use the authenticator information to define acceptable activity differently for multiple classes of users or specific users.

The “protocol analysis” performed by stateful protocol analysis methods usually includes reasonableness checks for individual commands, such as minimum and maximum lengths for arguments. If a command typically has a username argument, and usernames have a maximum length of 20 characters, then an argument with a length of 1000 characters is suspicious. If the large argument contains binary data, then it is even more suspicious.

Stateful protocol analysis methods use protocol models, which are usually based primarily on standards from software vendors and standards bodies (e.g., Internet Engineering Task Force [IETF] Request for Comments [RFC]). The protocol models also typically take into account variances in each protocol's implementation. Many standards are not exhaustively complete, and vendors may violate standards or add proprietary features; all of these situations can cause variations among implementations. For proprietary protocols, complete details about the protocols are often not available, making it difficult for IDPS technologies to perform comprehensive, accurate analysis. Also, as protocols are revised and vendors alter their protocol implementations, IDPS protocol models need to be updated to reflect those changes.

The primary drawback to stateful protocol analysis methods is that they are very resource-intensive because of the complexity of the analysis and the overhead involved in performing state tracking for many simultaneous sessions. Another problem is that stateful protocol analysis methods cannot detect attacks that do not violate the characteristics of generally acceptable protocol behavior, such as performing many benign actions in a short period of time to cause a denial of service. Yet another problem is that the protocol model used by an IDPS might conflict with the way the protocol is implemented in particular versions of specific applications and operating systems, or how different client and server implementations of the protocol interact.

1.1.2 IDPS Components

The typical components in an IDPS solution are:

- Sensor or Agent. Sensors and agents monitor and analyze activity. The term "sensor" is typically used for IDPSs that monitor networks, and the term "agent" for IDPS technologies that monitor only a single host.
- Management Server. A management server is a device that receives information from sensors or agents and manages them. Some management servers perform analysis on the received information and can identify incidents that the individual sensors or agents cannot. Matching event information from multiple sensors or agents, such as finding events triggered by the same IP address, is known as correlation. Some small IDPS deployments do not use any management servers. In larger IDPS deployments, there are often multiple management servers, sometimes in tiers.
- Database Server. A database server is a repository for event information recorded by sensors, agents, and management servers. Many IDPSs support the use of database servers.
- Console. A console is a program that provides an interface for the IDPS's users and administrators. Console software is typically installed onto standard desktop or laptop computers. Some consoles are used for IDPS administration only, such as configuring sensors or agents and applying software updates, while other consoles are used strictly for monitoring and analysis. Some IDPS consoles provide both administration and monitoring capabilities.

IDPS components can be connected to each other through regular networks or a separate network designed for security software management known as a management network. If a management network is used, each sensor or agent host has an additional network interface known as a management interface that connects to the management network, and the hosts are configured so that they cannot pass any traffic between management interfaces and other network interfaces. The management servers, database servers, and consoles are attached to the management network only. This architecture effectively isolates the management network from the production networks, concealing the IDPS from attackers and ensuring that the IDPS has adequate bandwidth to function under adverse conditions. If an IDPS is deployed without a separate management network, a way of improving IDPS security is to create a virtual management network using a virtual local area network (VLAN) within the standard networks. Using a VLAN provides protection for IDPS communications, but not as much protection as a separate management network.

1.1.3 IDPS Security Capabilities

IDPS technologies typically offer extensive, broad detection capabilities. Most products use a combination of detection techniques, which generally supports more accurate detection and more flexibility in tuning and customization. The types of events detected and the typical accuracy of detection vary greatly depending on the type of IDPS technology. Most IDPSs require at least some tuning and customization to improve their detection accuracy, usability, and effectiveness. Examples of tuning and customization capabilities are as follows:

- Thresholds. A threshold is a value that sets the limit between normal and abnormal behavior. Thresholds usually specify a maximum acceptable level, such as 5 failed connection attempts in 60 seconds, or 100 characters for a filename length.
- Blacklists and Whitelists. A blacklist is a list of discrete entities, such as hosts, Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers, Internet Control Message Protocol (ICMP) types and codes, applications, usernames, Uniform Resource Locators (URLs), filenames, or file extensions, that have been previously determined to be associated with malicious activity. Blacklists allow IDPSs to block activity that is highly likely to be malicious. 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.
- 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 and setting a default priority or severity level. Some products can 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. Viewing the code can help analysts to determine why

particular alerts were generated, helping to validate alerts and identify false positives. The ability to edit detection-related code and write new code (e.g., new signatures) is necessary to fully customize certain types of detection capabilities.

Most IDPSs offer multiple prevention capabilities; the specific capabilities vary by IDPS technology type. IDPSs usually allow administrators to specify the prevention capability configuration for each type of alert. This usually includes enabling or disabling prevention, as well as specifying which type of prevention capability should be used. Some IDPS technologies offer information gathering capabilities, such as collecting information on hosts or networks from observed activity. Examples include identifying hosts and the operating systems and applications that they use, and identifying general characteristics of the network.

1.2 Types of IDPS Technologies

There are many types of IDPS technologies. For the purposes of this document, they are divided into the following four groups based on the type of events that they monitor and the ways in which they are deployed:

- Network-Based, which monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. It can identify many different types of events of interest. It is most commonly deployed at a boundary between networks, such as in proximity to border firewalls or routers, remote access servers, and wireless networks.
- Wireless, which monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity involving the protocols themselves. It cannot identify suspicious activity in the application or higher-layer network protocols (e.g., TCP, UDP) that the wireless network traffic is transferring. It is most commonly deployed within range of an organization's wireless network to monitor it, but can also be deployed to locations where unauthorized wireless networking could be occurring.
- Network Behavior Analysis (NBA), which examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS) attacks, certain forms of malware (e.g., worms, backdoors), and policy violations (e.g., a client system providing unauthorized network services to other systems). NBA systems are most often deployed to monitor flows on an organization's internal networks, and are also sometimes deployed where they can monitor flows between an organization's networks and external networks (e.g., the Internet, business partners' networks).
- Host-Based, which monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Examples of the types of characteristics a host-based IDPS might monitor are network traffic (only for that host), system logs, running processes, application activity, file access and modification, and system and application configuration changes. Host-based IDPSs are most commonly deployed on critical hosts such as publicly accessible servers and servers containing sensitive information.

This portion of the chapter discusses each of these four groups in more detail. For each group, it gives a general overview and then discusses the IDPS's security capabilities and limitations in more detail.

1.2.1 Network-Based IDPS

A network-based IDPS monitors and analyzes network traffic for particular network segments or devices to identify suspicious activity. Network-based IDPSs are most often deployed at the division between networks. The IDPS network interface cards that will be performing monitoring are placed into promiscuous mode so that they accept all packets that they see, regardless of their intended destinations. Network-based IDPSs typically perform most of their analysis at the application layer (e.g., Hypertext Transfer Protocol [HTTP], Simple Mail Transfer Protocol [SMTP], Domain Name System [DNS]). They also analyze activity at the transport (e.g., TCP, UDP) and network (e.g., IPv4) layers to identify attacks at those layers and facilitate application layer analysis. Some network-based IDPSs also perform limited analysis at the hardware layer (e.g., Address Resolution Protocol [ARP]).

Network-based IDPS sensors can be deployed in one of two modes: inline or passive. An inline sensor is deployed so that the traffic it monitors passes through it. Some inline sensors are hybrid firewall/IDPS devices. The primary motivation for deploying sensors inline is to stop attacks by blocking traffic. A passive sensor is deployed so that it monitors a copy of the actual traffic; no traffic passes through the sensor. Passive sensors can monitor traffic through various methods, including a switch spanning port, which can see all traffic going through the switch; a network tap, which is a direct connection between a sensor and the physical network media itself, such as a fiber optic cable; and an IDS load balancer, which is a device that aggregates and directs traffic to monitoring systems. Most techniques for having a sensor prevent intrusions require that the sensor be deployed in inline mode. Passive techniques typically provide no reliable way for a sensor to block traffic. In some cases, a passive sensor can place packets onto a network to attempt to disrupt a connection, but such methods are generally less effective than inline methods.

IP addresses are normally not assigned to the sensor network interfaces used to monitor traffic, except for network interfaces also used for IDPS management. Operating a sensor without IP addresses assigned to its monitoring interfaces is known as stealth mode. It improves the security of the sensors because it conceals them and prevents other hosts from initiating connections to them. However, attackers may be able to identify the existence of a sensor and determine which product is in use by analyzing the characteristics of its prevention actions. Such analysis might include monitoring protected networks and determining which scan patterns trigger particular responses and what values are set in certain packet header fields.

Security Capabilities

Network-based IDPSs typically offer extensive and broad detection capabilities. Most use a combination of signature-based, anomaly-based, and stateful protocol analysis detection techniques. These techniques are usually tightly interwoven; for example, a stateful protocol analysis engine might parse activity into requests and responses, each of which is examined for anomalies and compared to signatures of known bad activity.

The types of events most commonly detected by network-based IDPS sensors include application, transport, and network layer reconnaissance and attacks. Many sensors can also detect unexpected application services, such as tunneled protocols, backdoors, and hosts running unauthorized applications. Also, some types of security policy violations can be detected by sensors that allow administrators to specify the characteristics of activity that should not be permitted, such as TCP or UDP port numbers, IP addresses, and Web site names. Some sensors can also monitor the initial negotiation conducted when establishing encrypted communications to identify client or server software that has known vulnerabilities or is misconfigured. Examples include secure shell (SSH), Transport Layer Security (TLS), and IP Security (IPsec).

Historically, network-based IDPSs have been associated with high rates of false positives and false negatives. These rates can only be reduced somewhat because of the complexity of the activities being monitored. A single sensor may monitor traffic involving hundreds or thousands of internal and external hosts, which run a wide variety of frequently-changing applications and operating systems (OSs). A sensor cannot understand everything it sees. Another common problem with detection accuracy is that the IDPS typically requires considerable tuning and customization to take into account the characteristics of the monitored environment. Also, security controls that alter network activity, such as firewalls and proxy servers, could cause additional difficulties for sensors by changing the characteristics of traffic.

Some network-based IDPSs can collect limited information on hosts and their network activity. Examples of this are a list of hosts on the organization's network, the operating system versions and application versions used by these hosts, and general information about network characteristics, such as the number of hops between devices. This information can be used by some IDPSs to improve detection accuracy. For example, an IDPS might allow administrators to specify the IP addresses used by the organization's Web servers, mail servers, and other common types of hosts, and also specify the types of services provided by each host (e.g., the Web server application type and version run by each Web server). This allows the IDPS to better prioritize alerts; for example, an alert for an Apache attack directed at an Apache Web server would have a higher priority than the same attack directed at a different type of Web server. Some network-based IDPSs can also import the results of vulnerability scans and use them to determine which attacks would likely be successful if not blocked. This allows the IDPS to make better decisions on prevention actions and prioritize alerts more accurately.

Network-based IDPS sensors offer various prevention capabilities. A passive sensor can attempt to end an existing TCP session by sending TCP reset packets to both endpoints, to make it appear to each endpoint that the other is trying to end the connection. However, this technique often cannot be performed in time to stop an attack and can only be used for TCP; other, newer prevention capabilities are more effective. Inline sensors can perform inline firewalling, throttle bandwidth usage, and alter malicious content. Both passive and inline sensors can reconfigure other network security devices to block malicious activity or route it elsewhere, and some sensors can run a script or program when certain malicious activity is detected to trigger custom actions.

Technology Limitations

Although network-based IDPSs offer extensive detection capabilities, they do have some significant limitations. Network-based IDPSs cannot detect attacks within encrypted traffic, including virtual private network (VPN) connections, HTTP over SSL (HTTPS), and SSH sessions. To ensure that sufficient analysis is performed on payloads within encrypted traffic, IDPSs can be deployed to analyze the payloads before they are encrypted or after they are decrypted. Examples include placing network-based IDPS sensors to monitor decrypted traffic and using host-based IDPS software to monitor activity within the source or destination host.

Network-based IDPSs may be unable to perform full analysis under high loads. This could cause some attacks to go undetected, especially if stateful protocol analysis methods are in use. For inline IDPS sensors, dropping packets also causes disruptions in network availability, and delays in processing packets could cause unacceptable latency. To avoid this, some inline IDPS sensors can recognize high load conditions and either pass certain types of traffic through the sensor without performing full analysis or drop low-priority traffic. Sensors may also provide better performance under high loads if they use specialized hardware (e.g., high-bandwidth network cards) or recompile components of their software to incorporate settings and other customizations made by administrators.

IDPS sensors are susceptible to various types of attacks. Attackers can generate large volumes of traffic, such as DDoS attacks, and other anomalous activity (e.g., unusually fragmented packets) to exhaust a sensor's resources or cause it to crash. Another attack technique, known as blinding, generates traffic that is likely to trigger many IDPS alerts quickly. In many cases, the blinding traffic is not intended to actually attack any targets. An attacker runs the "real" attack separately at the same time as the blinding traffic, hoping that the blinding traffic will either cause the IDPS to fail in some way or cause the alerts for the real attack to go unnoticed. Many sensors can recognize common attacks against them, alert administrators to the attack, and then ignore the rest of the activity.

1.2.2 Wireless IDPS

A wireless IDPS monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity involving those protocols. Wireless IDPSs are most often used for monitoring wireless local area networks (WLAN). WLANs are typically used by devices within a fairly limited range, such as an office building or corporate campus, and are implemented as extensions to existing wired local area networks (LAN) to provide enhanced user mobility.

Most WLANs use the Institute of Electrical and Electronics Engineers (IEEE) 802.11 family of WLAN standards. IEEE 802.11 WLANs have two fundamental architectural components: a station (STA), which is a wireless endpoint device (e.g., laptop computer, personal digital assistant [PDA]), and an access point (AP), which logically connects STAs with an organization's wired network infrastructure or other network. Some WLANs also use wireless switches, which act as intermediaries between APs and the wired network. A network based on STAs and APs is configured in infrastructure mode; a network that does not use an AP, in which STAs connect directly to each other, is configured in ad hoc mode. Nearly all organization

WLANs use infrastructure mode. Each AP in a WLAN has a name assigned to it called a service set identifier (SSID). The SSID allows STAs to distinguish one WLAN from another.

The typical components in a wireless IDPS are the same as a network-based IDPS, other than sensors. Wireless sensors function very differently because of the complexities of monitoring wireless communications. Unlike a network-based IDPS, which can see all packets on the networks it monitors, a wireless IDPS works by sampling traffic. There are two frequency bands to monitor (2.4 gigahertz [GHz] and 5 GHz), and each band is separated into channels. A sensor cannot monitor all traffic on a band simultaneously—it has to monitor a single channel at a time. The longer a single channel is monitored, the more likely it is that the sensor will miss malicious activity occurring on other channels. To avoid this, sensors typically change channels frequently, which is known as channel scanning. To reduce channel scanning, specialized sensors are available that use several radios and high-power antennas. Because of their higher sensitivities, the high-power antennas also have a larger monitoring range than regular antennas. Some implementations coordinate scanning patterns among sensors with overlapping ranges so that each sensor needs to monitor fewer channels.

Wireless sensors are available in several forms. A dedicated sensor is usually passive, performing wireless IDPS functions but not passing traffic from source to destination. Dedicated sensors may be designed for fixed or mobile deployment, with mobile sensors used primarily for auditing and incident handling purposes (e.g., to locate rogue wireless devices). Sensor software is also available bundled with APs and wireless switches. Some vendors also have host-based wireless IDPS sensor software that can be installed on STAs, such as laptops. The sensor software detects STA misconfigurations and attacks within range of the STAs. The sensor software may also be able to enforce security policies on the STAs, such as limiting access to wireless interfaces.

If an organization uses WLANs, it most often deploys wireless sensors to monitor the radio frequency (RF) range of the organization’s WLANs, which often includes mobile components such as laptops and PDAs. Many organizations also use sensors to monitor areas of their facilities where there should be no WLAN activity, as well as channels and bands that the organization’s WLANs should not use, as a way of detecting rogue devices.

Security Capabilities

Wireless IDPSs can detect attacks, misconfigurations, and policy violations at the WLAN protocol level, primarily examining IEEE 802.11 protocol communication. Wireless IDPSs do not examine communications at higher levels (e.g., IP addresses, application payloads). Some products perform only simple signature-based detection, while others use a combination of signature-based, anomaly-based, and stateful protocol analysis detection techniques. The types of events most commonly detected by wireless IDPS sensors include unauthorized WLANs and WLAN devices and poorly secured WLAN devices (e.g., misconfigured WLAN settings). Wireless IDPSs can also detect unusual WLAN usage patterns, which could indicate a device compromise or unauthorized use of the WLAN, and the use of wireless network scanners. Denial of service conditions, including logical attacks (e.g., overloading APs with large numbers of messages) and physical attacks (e.g., emitting electromagnetic energy on the WLAN’s frequencies to make the WLAN unusable), can also be detected by wireless IDPS. Some wireless IDPSs can also detect a WLAN device that attempts to spoof the identity of another device.

Most wireless IDPS sensors can identify the physical location of a wireless device by using triangulation—estimating the device's approximate distance from multiple sensors by the strength of the device's signal received by each sensor, then calculating the physical location at which the device would be the estimated distance from each sensor. Handheld IDPS sensors can also be used to pinpoint a device's location, particularly if fixed sensors do not offer triangulation capabilities or if the device is moving.

Compared to other forms of IDPS, wireless IDPS is generally more accurate; this is largely due to its narrow focus. False positives are most likely to be caused by anomaly-based detection methods, especially if threshold values are not properly maintained. Although many alerts might occur based on benign activity, such as another organization's WLAN being within range of the organization's WLANs, these alerts are not truly false positives because they are accurately detecting an unknown WLAN.

Wireless IDPS technologies usually require some tuning and customization to improve their detection accuracy. The main effort is in specifying which WLANs, APs, and STAs are authorized, and in entering the policy characteristics into the wireless IDPS software. Because wireless IDPSs are only examining wireless network protocols, not higher level protocols (e.g., applications), there are generally not a large number of alert types, and consequently not many customizations or tunings available.

Wireless IDPS sensors offer two types of intrusion prevention capabilities. Some sensors can terminate connections through the air, typically by sending messages to the endpoints, telling them to deassociate the current session, then refusing to permit a new connection to be established. Another prevention method is for a sensor to instruct a switch on the wired network to block network activity involving a particular device based on the device's media access control (MAC) address or switch port. However, this technique is only effective for blocking the device's communications on the wired network, not the wireless network. An important consideration when choosing prevention capabilities is the effect that prevention actions can have on sensor monitoring. For example, if a sensor is transmitting signals to terminate connections, it may not be able to perform channel scanning to monitor other communications until it has completed the prevention action. To mitigate this, some sensors have two radios—one for monitoring and detection, and another for performing prevention actions.

Technology Limitations

Although wireless IDPSs offer robust detection capabilities, they do have some significant limitations. One problem with some wireless IDPS sensors is the use of evasion techniques, particularly against sensor channel scanning schemes. One example is performing attacks in very short bursts on channels that are not currently being monitored. An attacker could also launch attacks on two channels at the same time. If the sensor detects the first attack, it cannot detect the second attack unless it scans away from the channel of the first attack.

Wireless IDPS sensors are also susceptible to attack. The same denial of service attacks (both logical and physical) that attempt to disrupt WLANs can also disrupt sensor functions. Sensors are also often particularly susceptible to physical attack because they are usually located in hallways, conference rooms, and other open areas. Some sensors have anti-tamper features, such as being designed to look like fire alarms or regular APs, that can reduce the likelihood that they will be attacked. All sensors are susceptible to physical attacks such as jamming that disrupt RF; there is no defense against such attacks other than to establish a physical perimeter around the facility so that attackers cannot get close enough to the WLAN to jam it.

It is also important to realize that wireless IDPSs cannot detect certain types of attacks against wireless networks. An attacker can passively monitor wireless traffic, which is not detectable by wireless IDPSs. If weak security methods are being used (e.g., Wired Equivalent Privacy [WEP]), the attacker can then perform offline processing of that collected traffic to find the encryption key used to provide security for the wireless traffic. With this key, the attacker can decrypt the traffic that was already collected, as well as any other traffic collected from the same WLAN. Wireless IDPSs cannot fully compensate for the use of insecure wireless networking protocols.

1.2.3 Network Behavior Analysis (NBA) System

A network behavior analysis (NBA) system examines network traffic or statistics on traffic to identify unusual traffic flows, such as DDoS attacks, certain forms of malware (e.g., worms, backdoors), and policy violations (e.g., a client system providing network services to other systems). Historically, NBA systems have been known by many names, including network behavior anomaly detection (NBAD) software, network behavior analysis and response software, and network anomaly detection software. NBA solutions usually have sensors and consoles, with some products also offering management servers (which are sometimes called analyzers).

Some sensors are similar to network-based IDPS sensors in that they sniff packets to monitor network activity on one or a few network segments. These sensors may be active or passive and are placed similarly to network-based IDS sensors—at the boundaries between networks, using the same connection methods. Other NBA sensors do not monitor the networks directly, but instead rely on network flow information provided by routers and other networking devices. Flow refers to a particular communication session occurring between hosts. Typical flow data includes source and destination IP addresses, source and destination TCP or UDP ports or ICMP types and codes, the number of packets and number of bytes transmitted in the session, and timestamps for the start and end of the session.

Security Capabilities

NBA technologies typically can detect several types of malicious activity. Most products use primarily anomaly-based detection, along with some stateful protocol analysis techniques. Most NBA technologies offer no signature-based detection capability, other than allowing administrators to manually set up custom filters that are essentially signatures to detect or stop specific attacks. The types of events most commonly detected by NBA sensors include network-based denial of service attacks, network scanning, worms, the use of unexpected application services, and policy violations (for example, a host attempting to contact another host with which it has no legitimate reason to communicate). Most NBA sensors can reconstruct a series of observed events to determine the origin of an attack. For example, if worms infect a network, NBA sensors can analyze the worm's flows and find the host on the organization's network that first transmitted the worm.

Because NBA sensors work primarily by detecting significant deviations from normal behavior, they are most accurate at detecting attacks that generate large amounts of network activity in a short period of time (e.g., DDoS attacks) and attacks that have unusual flow patterns (e.g., worms spreading among hosts). NBA sensors are less accurate at detecting small-scale attacks, particularly if they are conducted slowly. Because NBA technologies use primarily anomaly-based detection methods, they cannot detect many attacks until they reach a point where their activity is significantly different from what is expected. The point during the attack at which the NBA software detects it may vary considerably based on an NBA product's configuration. By configuring sensors to be more sensitive to anomalous activity, alerts will be generated more quickly when attacks occur, but more false positives are also likely to be triggered. Conversely, if sensors are configured to be less sensitive to anomalous activity, there will be fewer false positives, but alerts will be generated more slowly, allowing attacks to occur for longer periods of time. False positives can also be caused by benign changes in the environment. For example, if a new service is added to a host and hosts start using it, an NBA sensor is likely to detect this as anomalous.

NBA technologies rely primarily on observing network traffic and developing baselines of expected flows and inventories of host characteristics. NBA products automatically update their baselines on an ongoing basis. As a result, typically there is not much tuning or customization to be done. Administrators might adjust thresholds periodically (e.g., how much additional bandwidth usage should trigger an alert) to take into account changes to the environment.

A few NBA products offer limited signature-based detection capabilities. The supported signatures tend to be very simple, primarily looking for particular values in certain IP, TCP, UDP, or ICMP header fields. This capability is most helpful for inline NBA sensors because they can use the signatures to find and block attacks that a firewall or router might not be capable of blocking. However, even without a signature capability, an inline NBA sensor might be able to detect and block the attack because of its flow patterns.

NBA technologies offer extensive information gathering capabilities, because knowledge of the characteristics of the organization's hosts is needed for most of the NBA product's detection techniques. NBA sensors can automatically create and maintain lists of hosts communicating on the organization's monitored networks. They can monitor port usage, perform passive fingerprinting, and use other techniques to gather detailed information on the hosts. Information typically collected for each host includes IP address, OS, the network services the host provides, and the nature of the host's communications with other hosts. NBA sensors constantly monitor network activity for changes to this information. Additional information on each host's flows is also collected on an ongoing basis.

NBA sensors offer various intrusion prevention capabilities, including sending TCP reset packets to endpoints, performing inline firewalling, and reconfiguring other network security devices. Most NBA system implementations use prevention capabilities in a limited fashion or not at all because of false positives; erroneously blocking a single flow could cause major disruptions in network communications. Prevention capabilities are most often used for NBA sensors when blocking a specific known attack, such as a new worm.

Technology Limitations

NBA technologies have significant limitations. An important limitation is the delay in detecting attacks. Some delay is inherent in anomaly detection methods that are based on deviations from a baseline, such as increased bandwidth usage or additional connection attempts. However, NBA technologies often have additional delay caused by their data sources, especially when they rely on flow data from routers and other network devices. This data is often transferred to the NBA system in batches, as frequently as every minute or two, often much less frequently. Because of this delay, attacks that occur quickly, such as malware infestations and denial of service (DoS) attacks, may not be detected until they have already disrupted or damaged systems.

This delay can be avoided by using sensors that do their own packet captures and analysis instead of relying on flow data from other devices. However, performing packet captures and analysis is much more resource-intensive than analyzing flow data. A single sensor can analyze flow data from many networks or perform direct monitoring (packet captures) itself generally for a few networks at most. More sensors may be needed to do direct monitoring instead of using flow data.

1.2.4 Host-Based IDPS

A host-based IDPS monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Examples of the types of host characteristics a host-based IDPS might monitor are wired and wireless network traffic, system logs, running processes, file access and modification, and system and application configuration changes. Most host-based IDPSs have detection software known as agents installed on the hosts of interest. Each agent monitors activity on a single host and may perform prevention actions. Some agents monitor a single specific application service—for example, a Web server program; these agents are also known as application-based IDPSs.

Host-based IDPS agents are most commonly deployed to critical hosts, such as publicly accessible servers and servers containing sensitive information, although they can be deployed to other types of hosts as well. Some organizations use agents primarily to analyze activity that cannot be monitored by other security controls. For example, network-based IDPS sensors cannot analyze the activity within encrypted network communications, but host-based IDPS agents installed on endpoints can see the unencrypted activity. The network architecture for host-based IDPS deployments is typically simple. Because the agents are deployed to existing hosts on the organization's networks, the components usually communicate over those networks instead of using a separate management network.

To provide intrusion prevention capabilities, most IDPS agents alter the internal architecture of hosts. This is typically done through a shim, which is a layer of code placed between existing layers of code. A shim intercepts data at a point where it would normally be passed from one piece of code to another. The shim can then analyze the data and determine whether or not it should be allowed or denied. Host-based IDPS agents may use shims for several types of resources, including network traffic, filesystem activity, system calls, Windows registry activity, and common applications (e.g., email, Web). Some agents monitor activity without using shims, or they analyze artifacts of activity, such as log entries and file modifications. Although these methods are less intrusive to the host, these methods are also generally less effective at detecting attacks and often cannot perform prevention actions.

Security Capabilities

Most host-based IDPSs can detect several types of malicious activity. They often use a combination of signature-based detection techniques to identify known attacks, and anomaly-based detection techniques with policies or rulesets to identify previously unknown attacks.

The types of events detected by host-based IDPSs vary considerably based primarily on the detection techniques that they use. Some host-based IDPS products offer several of these detection techniques, while others focus on a few or one. Specific techniques commonly used in host-based IDPSs include the following:

- Code Analysis. Agents might analyze attempts to execute malicious code. One technique is executing code in a virtual environment or sandbox to analyze its behavior and compare it to profiles of known good and bad behavior. Another technique is looking for the typical characteristics of stack and heap buffer overflow exploits, such as certain sequences of instructions and attempts to access portions of memory not allocated to the process. System call monitoring is another common technique; it involves knowing which applications and processes should be performing certain actions.
- Network Traffic Analysis. This is often similar to what a network-based IDPS does. Some products can also analyze wireless traffic. Another capability of traffic analysis is that the agent can extract files sent by applications such as email, Web, and peer-to-peer file sharing, which can then be checked for malware.
- Network Traffic Filtering. Agents often include a host-based firewall that can restrict incoming and outgoing traffic for each application on the system, preventing unauthorized access and acceptable use policy violations (e.g., use of inappropriate external services).
- Filesystem Monitoring. Filesystem monitoring can be performed using several different techniques. File integrity checking involves generating cryptographic checksums for critical files and comparing them to reference values to identify which files have been changed. File attribute checking is the process of checking critical files' security attributes, such as ownership and permissions, for changes. Both file integrity checking and file attribute checking are reactive, detecting attacks only after they have occurred. Some agents have more proactive capabilities, such as monitoring file access attempts,

comparing each attempt to an access control policy, and preventing attempts that violate policy.

- Log Analysis. Some agents can monitor and analyze OS and application logs to identify malicious activity. These logs may contain information on system operational events, audit records, and application operational events.

Because host-based IDPSs often have extensive knowledge of hosts' characteristics and configurations, an agent can often determine whether an attack would succeed if not stopped. Agents can use this knowledge to select prevention actions and to prioritize alerts.

Like any other IDPS technology, host-based IDPSs often cause false positives and false negatives. However, the accuracy of detection is more challenging for host-based IDPSs because they detect events but do not have knowledge of the context under which the events occurred. For example, a new application may be installed—this could be done by malicious activity or done as part of normal host operation. The event's benign or malicious nature cannot be determined without additional context. Host-based IDPSs that use combinations of several detection techniques generally should achieve more accurate detection than products that use one or a few techniques. Because each technique can monitor different aspects of a host, using more techniques allows agents to have a more complete picture of the events, including additional context.

Host-based IDPSs usually require considerable tuning and customization. For example, many rely on observing host activity and developing profiles of expected behavior. Others need to be configured with detailed policies that define exactly how each application on a host should behave. As the host environment changes, policies need to be updated to take those changes into account. Some products permit multiple policies to be configured on a host for multiple environments; this is most helpful for hosts that function in multiple environments, such as a laptop used both within an organization and from external locations.

Host-based IDPS agents offer various intrusion prevention capabilities, based on the detection techniques they use. For example, code analysis techniques can prevent malicious code from being executed, and network traffic analysis techniques can stop incoming traffic from being processed by the host and can prevent malicious files from being placed on the host. Network traffic filtering techniques can block unwanted communications. Filesystem monitoring can prevent files from being accessed, modified, replaced, or deleted, which could stop malware installation, including Trojan horses and rootkits, as well as other attacks involving inappropriate file access. Other host-based IDPS detection techniques, such as log analysis, network configuration monitoring, and file integrity and attribute checking, generally do not support prevention actions because they identify events after they have occurred.

Technology Limitations

Host-based IDPSs have some significant limitations. Although agents generate alerts on a real-time basis for most detection techniques, some techniques are used periodically to identify events that have already happened. Such techniques might only be applied hourly or even just a few times a day, causing significant delay in identifying certain events. Also, many host-based

IDPSs are intended to forward their alert data to the management servers on a periodic basis, such as every 15 to 60 minutes, to reduce overhead. This can cause delays in initiating response actions, which especially increases the impact of incidents that spread quickly, such as malware infestations. Host-based IDPSs can consume considerable resources on the hosts that they protect, particularly if they use several detection techniques and if they use shims. Host-based IDPSs can also cause conflicts with existing security controls, such as personal firewalls, particularly if those controls also use shims to intercept host activity.

1.3 Using and Integrating Multiple IDPS Technologies

The four primary types of IDPS technologies—network-based, wireless, NBA, and host-based—each offer fundamentally different capabilities. Each technology type offers benefits over the other, such as detecting some attacks that the others cannot, detecting some attacks more accurately, and functioning without significantly impacting the performance of the protected hosts. Accordingly, using multiple types of IDPS technologies can achieve more comprehensive and accurate detection and prevention of malicious activity. For most environments, a combination of network-based and host-based IDPSs is needed at a minimum. Wireless IDPSs may also be needed if WLAN security or rogue WLAN detection is a concern. NBA products can also be deployed to achieve stronger detection capabilities for DoS attacks, worms, and other threats that cause anomalous network flows.

Some organizations also use multiple products of the same IDPS technology type to improve detection capabilities. Because each product detects some events that another product cannot, using multiple products can allow for more comprehensive detection. Also, having multiple products monitoring the same activity makes it easier for analysts to confirm the validity of alerts and identify false positives, and also provides redundancy.

1.3.1 Product Integration

By default, different IDPS products function completely independently of each other. This has some benefits, such as minimizing the impact that a failure or compromise of one IDPS product has on other IDPS products. However, if the products are not integrated in any way, the effectiveness of the entire IDPS implementation may be somewhat limited. Data cannot be shared by the products, and extra effort will be needed to monitor and manage multiple sets of products. IDPS products can be directly or indirectly integrated.

Direct IDPS integration involves one product feeding information to another product. Direct integration is most often performed when an organization uses multiple IDPS products from a single vendor. For example, a network-based IDPS sensor might use host-based IDPS data to determine if an attack detected by the network-based IDPS sensor was successful, and a network-based IDPS could provide network flow information to an NBA sensor. This information can improve detection accuracy, speed the analysis process, and help prioritize threats. The primary disadvantage of using a fully integrated solution is that a failure or compromise could endanger all the IDPS technologies that are part of it.

Indirect IDPS integration usually involves many IDPS products sending their data to security information and event management (SIEM) software. SIEM software is designed to import information from security-related logs and correlate events among them. Log types commonly supported by SIEM software include IDPSs, firewalls, antivirus software, and other security software; OSs (e.g., audit logs); application servers (e.g., Web servers, email servers); and even physical security devices such as badge readers. SIEM software generally works by receiving copies of the logs from the logging hosts over secure network channels, converting the log data into standard fields and values (known as normalization), then identifying related events by matching IP addresses, timestamps, usernames, and other characteristics. SIEM products can identify malicious activity such as attacks and malware infections, as well as misuse and inappropriate usage of systems and networks. Some SIEM software can also initiate prevention responses for designated events.

SIEM software can supplement IDPSs. For example, SIEM software can correlate events logged by different technologies. This can identify incidents that a single source could not, as well as collecting information related to an event in a single place to make analysis more efficient. However, SIEM software also has some significant limitations. There is often a considerable delay between the time an event begins and the time the SIEM sees the corresponding log data, since log data is often transferred in batch mode to conserve resources. Resource consumption is also limited by SIEM products transferring only some event data from the original sources.

An alternative to using SIEM software for centralized logging is to use a solution based primarily on the syslog protocol. Syslog provides a simple framework for log generation, storage, and transfer that any IDPS could use if designed to do so. Some IDPSs offer features that allow their log formats to be converted to syslog format. Syslog is very flexible for log sources, because each syslog entry contains a content field into which logging sources can place information in any format. However, this flexibility makes analysis of the log data challenging. Each IDPS may use many different formats for its log messages, so a robust analysis program would need to be familiar with each format and be able to extract the meaning of the data within the fields of each format. It might not be feasible to understand the meaning of all log messages, so analysis might be limited to keyword and pattern searches.

1.3.2 Complementary Technologies

In addition to dedicated IDPS technologies, organizations typically have several other types of technologies that offer some IDPS capabilities and complement the primary IDPSs. For example, network forensic analysis tools (NFAT) focus primarily on collecting and analyzing wired network traffic. Unlike a network-based IDPS, which performs in-depth analysis and stores only the necessary network traffic, an NFAT typically stores most or all of the traffic that it sees, and then performs analysis on that stored traffic. Also, an NFAT can search payloads for keywords and other specific content, which IDPSs may not be able to do. However, an NFAT does not offer the intrusion detection capabilities that IDPSs do.

There are several types of tools for detecting malware, with the most commonly used being antivirus software. Types of malware that it can detect include viruses, worms, Trojan horses, malicious mobile code, and blended threats, as well as attacker tools such as keystroke loggers and backdoors. Antivirus software typically monitors critical OS components, filesystems, and application activity for signs of malware, and attempts to disinfect or quarantine files that contain malware. Another common tool is antispyware software, which detects both malware and non-malware forms of spyware, such as malicious mobile code and tracking cookies, and spyware installation techniques such as unauthorized Web browser plug-in installations. Malware detection tools usually offer much more robust malware detection capabilities than IDPSs.

Another tool that provides limited IDPS capabilities is a honeypot. Honeypots are hosts that have no authorized users other than the honeypot administrators because they serve no business function; all activity directed at them is considered suspicious. Attackers will scan and attack honeypots, giving administrators data on new trends and attack tools, particularly malware. However, honeypots are a supplement to, not a replacement for, other security controls such as intrusion detection and prevention systems. If honeypots are to be used by an organization, qualified incident handlers and intrusion detection analysts should manage them. The legality of honeypots has not been clearly established; therefore, organizations should carefully study the legal ramifications before planning any honeypot deployments.

References

1.1 K. Scarfone, P. Mell, Special Publication 800-94: Guide to Intrusion Detection and Prevention Systems (IDPS), National Institute of Standards and Technology (NIST) (2007)

Note that this chapter has been derived from reference 1.1.