National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-94

Guide to Intrusion Detection and Prevention Systems (IDPS)

Recommendations of the National Institute of Standards and Technology

Karen Scarfone
Peter Mell

NIST Special Publication 800-94

Guide to Intrusion Detection and Prevention Systems (IDPS)

Recommendations of the National Institute of Standards and Technology

Karen Scarfone
Peter Mell

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

February 2007

U.S. Department of Commerce

Carlos M. Gutierrez, Secretary

Technology Administration

Robert C. Cresanti, Under Secretary of Commerce for Technology

National Institute of Standards and Technology

William Jeffrey, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-94
Natl. Inst. Stand. Technol. Spec. Publ. 800-94, 127 pages (February 2007)

Acknowledgements

The authors, Karen Scarfone and Peter Mell of the National Institute of Standards and Technology (NIST), wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge John Connor, Tim Grance, Anoop Singhal, and Murugiah Souppaya of NIST; Michael Gerdes, Ralph Martins, Angela Orebaugh, and Mike Zeberlein of Booz Allen Hamilton; and Steve Sharma of Project Performance Corporation for their keen and insightful assistance throughout the development of the document. The authors particularly want to thank Rebecca Bace of KSR for her careful review of the publication and for her work on the predecessor publication, NIST Special Publication 800-31, *Intrusion Detection Systems*. The authors would also like to express their thanks to security experts Andrew Balinsky (Cisco Systems), Anton Chuvakin (LogLogic), Jay Ennis (Network Chemistry), John Jerrim (Lancope), and Kerry Long (Center for Intrusion Monitoring and Protection, Army Research Laboratory), as well as representatives from the Department of State and Gartner, for their particularly valuable comments and suggestions. Additional acknowledgements will be added to the final version of the publication.

Trademarks

Table of Contents

List of Appendices

List of Figures

List of Tables

Executive Summary

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. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS)[1] are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content.

This publication describes the characteristics of IDPS technologies and provides recommendations for designing, implementing, configuring, securing, monitoring, and maintaining them. The types of IDPS technologies are differentiated primarily by the types of events that they monitor and the ways in which they are deployed. This publication discusses the following four types of IDPS technologies:

- **Network-Based**, which monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity
- **Wireless**, which monitors wireless network traffic and analyzes it to identify suspicious activity involving the wireless networking protocols themselves
- **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, and policy violations (e.g., a client system providing network services to other systems)
- **Host-Based,** which monitors the characteristics of a single host and the events occurring within that host for suspicious activity.

Implementing the following recommendations should facilitate more efficient and effective intrusion detection and prevention system use for Federal departments and agencies.

Organizations should ensure that all IDPS components are secured appropriately.

Securing IDPS components is very important because IDPSs are often targeted by attackers who want to prevent the IDPSs from detecting attacks or want to gain access to sensitive information in the IDPSs, such as host configurations and known vulnerabilities. IDPSs are composed of several types of components, including sensors or agents, management servers, database servers, user and administrator consoles, and management networks. All components' operating systems and applications should be kept fully up-to-date, and all software-based IDPS components should be hardened against threats. Specific

[1] 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 system* (*IDPS)* is used throughout the rest of this guide to refer to both IDS and IPS technologies.

protective actions of particular importance include creating separate accounts for each IDPS user and administrator, restricting network access to IDPS components, and ensuring that IDPS management communications are protected appropriately, such as encrypting them or transmitting them over a physically or logically separate network. Administrators should maintain the security of the IDPS components on an ongoing basis, including verifying that the components are functioning as desired, monitoring the components for security issues, performing regular vulnerability assessments, responding appropriately to vulnerabilities in the IDPS components, and testing and deploying IDPS updates. Administrators should also back up configuration settings periodically and before applying updates to ensure that existing settings are not inadvertently lost.

Organizations should consider using multiple types of IDPS technologies to achieve more comprehensive and accurate detection and prevention of malicious activity.

The four primary types of IDPS technologies—network-based, wireless, NBA, and host-based—each offer fundamentally different information gathering, logging, detection, and prevention capabilities. Each technology type offers benefits over the others, such as detecting some events that the others cannot and detecting some events with significantly greater accuracy than the other technologies. In many environments, a robust IDPS solution cannot be achieved without using multiple types of IDPS technologies. For most environments, a combination of network-based and host-based IDPS technologies is needed for an effective IDPS solution. Wireless IDPS technologies may also be needed if the organization determines that its wireless networks need additional monitoring or if the organization wants to ensure that rogue wireless networks are not in use in the organization's facilities. NBA technologies can also be deployed if organizations desire additional detection capabilities for denial of service attacks, worms, and other threats that NBAs are particularly well-suited to detecting. Organizations should consider the different capabilities of each technology type along with other cost-benefit information when selecting IDPS technologies.

Organizations planning to use multiple types of IDPS technologies or multiple products of the same IDPS technology type should consider whether or not the IDPSs should be integrated.

Direct IDPS integration most often occurs when an organization uses multiple IDPS products from a single vendor, by having a single console that can be used to manage and monitor the multiple products. Some products can also mutually share data, which can speed the analysis process and help users to better prioritize threats. A more limited form of direct IDPS integration is having one IDPS product provide data for another IDPS product (but no data sharing in the opposite direction). Indirect IDPS integration is usually performed with security information and event management (SIEM) software, which is designed to import information from various security-related logs and correlate events among them. SIEM software complements IDPS technologies in several ways, including correlating events logged by different technologies, displaying data from many event sources, and providing supporting information from other sources to help users verify the accuracy of IDPS alerts.

Before evaluating IDPS products, organizations should define the requirements that the products should meet.

Evaluators need to understand the characteristics of the organization's system and network environments, so that a compatible IDPS can be selected that can monitor the events of interest on the systems and/or networks. Evaluators should articulate the goals and objectives they wish to attain by using an IDPS, such as stopping common attacks, identifying misconfigured wireless network devices, and detecting misuse of the organization's system and network resources. Evaluators should also review their existing security policies, which serve as a specification for many of the features that the IDPS products need to provide. In addition, evaluators should understand whether or not the organization is subject to oversight

or review by another organization. If so, they should determine if that oversight authority requires IDPSs or other specific system security resources. Resource constraints should also be taken into consideration by evaluators. Evaluators also need to define specialized sets of requirements for the following:

- Security capabilities, including information gathering, logging, detection, and prevention
- Performance, including maximum capacity and performance features
- Management, including design and implementation (e.g., reliability, interoperability, scalability, product security), operation and maintenance (including software updates), and training, documentation, and technical support
- Life cycle costs, both initial and maintenance costs.

When evaluating IDPS products, organizations should consider using a combination of several sources of data on the products' characteristics and capabilities.

Common product data sources include test lab or real-world product testing, vendor-provided information, third-party product reviews, and previous IDPS experience from individuals within the organization and trusted individuals at other organizations. When using data from other parties, organizations should consider the fidelity of the data because it is often presented without an explanation of how it was generated. There are several major challenges in performing in-depth hands-on IDPS testing, such as the considerable resources needed and the lack of a standard test methodology and test suites, which often make it infeasible. However, limited IDPS testing is helpful for evaluating security requirements, performance, and operation and maintenance capabilities.

This page has been left blank intentionally.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Purpose and Scope

This publication seeks to assist organizations in understanding intrusion detection system (IDS) and intrusion prevention system (IPS) technologies and in designing, implementing, configuring, securing, monitoring, and maintaining intrusion detection and prevention systems (IDPS). It provides practical, real-world guidance for each of four classes of IDPS products: network-based, wireless, network behavior analysis, and host-based. The publication also provides an overview of complementary technologies that can detect intrusions, such as security information and event management software and network forensic analysis tools. It focuses on enterprise IDPS solutions, but most of the information in the publication is also applicable to standalone and small-scale IDPS deployments. This publication replaces NIST Special Publication 800-31, *Intrusion Detection Systems*.

1.3 Audience

This document has been created for computer security staff and program managers, computer security incident response teams (CSIRT), and system and network administrators who are responsible for managing or monitoring IDPS technologies. This document does not assume that the reader has previous experience with any IDPS technologies, but it does assume that the reader has experience with information security.

1.4 Document Structure

The remainder of this document is organized into the following nine major sections:

- Section 2 provides an introduction to the basic concepts of intrusion detection and prevention.
- Section 3 gives an overview of IDPS technologies, including typical components, general detection methodologies, and implementation and operation guidance.

- Sections 4 through 7 contain detailed discussions of particular categories of IDPS technologies:
  - Section 4: Network-based
  - Section 5: Wireless
  - Section 6: Network behavior analysis
  - Section 7: Host-based
- Section 8 discusses other technologies with IDPS capabilities.
- Section 9 provides recommendations for using and integrating multiple IDPS technologies within an enterprise.
- Section 10 gives guidance on IDPS product selection.

The document also contains appendices with supporting material. Appendices A and B contain a glossary and acronym list, respectively. Appendix C lists print resources and online tools and resources that may be useful references for gaining a better understanding of IDPSs. Appendix D contains an index for the publication.

2. Intrusion Detection and Prevention Principles

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. Incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization.

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. This section provides an overview of IDS and IPS technologies as a foundation for the rest of the publication. It first explains how IDS and IPS technologies can be used. Next, it describes the key functions that IDS and IPS technologies perform and the detection methodologies that they use. Finally, it provides an overview of the major classes of IDS and IPS technologies.

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 guide to refer to both IDS and IPS technologies.[2] Any exceptions are specifically noted.

2.1 Uses of IDPS Technologies

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 could then report the incident to security administrators, who could quickly initiate incident response actions to minimize the damage caused by the incident.[3] The IDPS could also log information that could be used by the incident handlers.[4] Many IDPSs can also be configured to recognize violations of security policies. For example, some IDPSs can be configured with firewall ruleset-like settings, allowing them to identify network traffic that violates the organization's security or acceptable use policies. Also, some IDPSs can monitor file transfers and identify ones that might be suspicious, such as copying a large database onto a user's laptop.

Many IDPSs can also identify reconnaissance activity, which may indicate that an attack is imminent. For example, some attack tools and forms of malware, particularly worms, perform reconnaissance activities such as host and port scans to identify targets for subsequent attacks. An IDPS might be able to block reconnaissance and notify security administrators, who can take actions if needed to alter other security controls to prevent related incidents. Because reconnaissance activity is so frequent on the Internet, reconnaissance detection is often performed primarily on protected internal networks.

---

2 This term is used for the purposes of this publication. It has not been widely used in the security community, and the reason for using it in this publication is strictly brevity, not to replace the well-established "IDS" and "IPS" terms.

3 If the IDPS had successfully prevented the attack, security administrators still might want to be notified of the attack. This is particularly important if the target has a known vulnerability that the attack could have exploited. Attackers could potentially use a different attack for the same vulnerability that the IDPS might not recognize.

4 A detailed discussion of incident response is outside the scope of this guide. For guidance on establishing an effective incident response capability, see NIST Special Publication (SP) 800-61, *Computer Security Incident Handling Guide*, which is available at http://csrc.nist.gov/publications/nistpubs/.

In addition to identifying incidents and supporting incident response efforts, organizations have found other uses for IDPSs, including the following:

- **Identifying security policy problems.** An IDPS can provide some degree of quality control for security policy implementation, such as duplicating firewall rulesets and alerting when it sees network traffic that should have been blocked by the firewall but was not because of a firewall configuration error.
- **Documenting the existing threat to an organization.** IDPSs log information about the threats that they detect. Understanding the frequency and characteristics of attacks against an organization's computing resources is helpful in identifying the appropriate security measures for protecting the resources. The information can also be used to educate management about the threats that the organization faces.
- **Deterring individuals from violating security policies.** If individuals are aware that their actions are being monitored by IDPS technologies for security policy violations, they may be less likely to commit such violations because of the risk of detection.

Because of the increasing dependence on information systems and the prevalence and potential impact of intrusions against those systems, IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

2.2 Key Functions of IDPS Technologies

There are many types of IDPS technologies, which are differentiated primarily by the types of events that they can recognize and the methodologies that they use to identify incidents. In addition to monitoring and analyzing events to identify undesirable activity, all types of IDPS technologies typically perform the following functions:

- **Recording information related to observed events.** Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management (SIEM) solutions, and enterprise management systems.
- **Notifying security administrators of important observed events.** This notification, known as an *alert*, occurs through any of several methods, including the following: e-mails, pages, messages on the IDPS user interface, Simple Network Management Protocol (SNMP) traps, syslog messages, and user-defined programs and scripts. A notification message typically includes only basic information regarding an event; administrators need to access the IDPS for additional information.
- **Producing reports.** Reports summarize the monitored events or provide details on particular events of interest.

Some IDPSs are also able to change their security profile when a new threat is detected. For example, an IDPS might be able to collect more detailed information for a particular session after malicious activity is detected within that session. An IDPS might also alter the settings for when certain alerts are triggered or what priority should be assigned to subsequent alerts after a particular threat is detected.

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

- **The IPS stops the attack itself.** Examples of how this could be done are as follows:

- Terminate the network connection or user session that is being used for the attack
- Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute
- Block all access to the targeted host, service, application, or other resource.

- **The IPS changes the security environment.** The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target, and altering a host-based firewall on a target to block incoming attacks. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.
- **The IPS changes the attack's content.** Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned 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.

Another common attribute of IDPS technologies is that they cannot provide completely accurate detection. When an IDPS incorrectly identifies benign activity as being malicious, a *false positive* has occurred. When an IDPS fails to identify malicious activity, a *false negative* has occurred. 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 offer features that 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 their attacks. For example, an attacker could encode text characters in a particular way, knowing that the target understands the encoding and hoping that any monitoring IDPSs do not. Most IDPS technologies can overcome common evasion techniques by duplicating special processing performed by the targets. If the IDPS can "see" the activity in the same way that the target would, then evasion techniques will generally be unsuccessful at hiding attacks.

2.3 Common Detection Methodologies

IDPS technologies use many methodologies to detect incidents. Sections 2.3.1 through 2.3.3 discuss the primary classes of detection methodologies: signature-based, anomaly-based, and stateful protocol analysis, respectively. Most IDPS technologies use multiple detection methodologies, either separately or integrated, to provide more broad and accurate detection.

2.3.1 Signature-Based Detection

A *signature* is a pattern that corresponds to a known threat. *Signature-based detection* is the process of comparing signatures against observed events to identify possible incidents.[5] Examples of signatures are as follows:

- A telnet attempt with a username of "root", which is a violation of an organization's security policy
- An e-mail 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 threats but largely ineffective at detecting previously unknown threats, threats disguised by the use of evasion techniques, and many variants of known threats. 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. Signature-based detection technologies have little understanding of many network or application protocols and cannot track and understand the state of complex communications. For example, they cannot pair a request with the corresponding response, such as knowing that a request to a Web server for a particular page generated a response status code of 403, meaning that the server refused to fill the request. They also lack the ability to remember previous requests when processing the current request. This limitation prevents signature-based detection methods from detecting attacks that comprise multiple events if none of the events contains a clear indication of an attack.

2.3.2 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 e-mails 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 threats. For example, suppose that a computer becomes infected with a new type of malware. The malware could consume the computer's processing resources, send large numbers of e-

[5] Signature-based detection is sometimes referred to as *misuse detection*, but this publication does not use that term because it implies that misuse is only detected using signatures, which is not true. Also, signature-based detection is considered by some parties to include stateful protocol analysis, as described in Section 2.3.3. For the purposes of this publication, signature-based detection is defined so as not to include stateful protocol analysis, but other publications may have different definitions.

mails, 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 (typically days, sometimes weeks) sometimes called a *training period*. Profiles for anomaly-based detection 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 slowly 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. Malicious activity might also be observed by an IDPS while it builds its initial profiles.

Inadvertently including malicious activity as part of a profile is a common problem with anomaly-based IDPS products. (In some cases, administrators can modify the profile to exclude activity in the profile that is known to be malicious.) Another problem with building profiles is that it can be very challenging in some cases to make them accurate, because computing activity can be 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 and trigger an alert. 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 why a particular alert was generated and to validate that an alert is accurate and not a false positive, because of the complexity of events and number of events that may have caused the alert to be generated.

2.3.3 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.[6] 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 finding 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.

[6] Some vendors use the term "deep packet inspection" to refer to performing some type of stateful protocol analysis, often combined with a firewall capability that can block communications determined to be malicious. This publication uses the term "stateful protocol analysis" because it is appropriate for analyzing both network-based and host-based activity, whereas "deep packet inspection" is an appropriate term for network-based activity only. Also, historically there has not been consensus in the security community as to the meaning of "deep packet inspection".

Stateful protocol analysis can identify unexpected sequences of commands, such as issuing the same command repeatedly or issuing a command without first issuing a command upon which it is dependent. Another state tracking feature of stateful protocol analysis is that for protocols that perform authentication, the IDPS can keep track of the authenticator used for each session, and record the authenticator used for suspicious activity. This is helpful when investigating an incident. 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 typically based primarily on protocol 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 in explaining the details of the protocol, which causes variations among implementations. Also, many vendors either violate standards or add proprietary features, some of which may replace features from the standards. For proprietary protocols, complete details about the protocols are often not available, making it difficult for IDPS technologies to perform comprehensive, accurate analysis. 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 serious 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.

2.4 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, virtual private network (VPN) servers, remote access servers, and wireless networks. Section 4 contains extensive information on network-based IDPS technologies.
- **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. More information on wireless IDPSs is presented in Section 5.

- **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 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). NBA products are discussed in more detail in Section 6.
- **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. Section 7 contains additional information on host-based IDPSs.

Some forms of IDPS are more mature than others because they have been in use much longer. Network-based IDPS and some forms of host-based IDPS have been commercially available for over ten years. Network behavior analysis software is a somewhat newer form of IDPS that evolved in part from products created primarily to detect DDoS attacks, and in part from products developed to monitor traffic flows on internal networks. Wireless technologies are a relatively new type of IDPS, developed in response to the popularity of wireless local area networks (WLAN) and the growing threats against WLANs and WLAN clients.

2.5 Summary

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. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

There are many types of IDPS technologies, which are differentiated primarily by the types of events that they can recognize and the methodologies that they use to identify possible incidents. This publication discusses the following four types of IDPS technologies:

- **Network-Based**, which monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity.
- **Wireless**, which monitors wireless network traffic and analyzes it to identify suspicious activity involving the wireless networking protocols themselves.
- **Network Behavior Analysis (NBA)**, which examines network traffic to identify threats that generate unusual traffic flows, such as DDoS attacks, scanning, and certain forms of malware.
- **Host-Based,** which monitors the characteristics of a single host and the events occurring within that host for suspicious activity.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's content.

IDPSs cannot provide completely accurate detection; they all generate false positives (incorrectly identifying benign activity as malicious) and false negatives (failing to identify malicious activity). Many organizations choose to tune IDPSs so that false negatives are decreased and false positives increased, which necessitates additional analysis resources to differentiate false positives from true malicious events. Most IDPSs also offer features that compensate for the use of common evasion techniques, which modify the format or timing of malicious activity to alter its appearance but not its effect, to attempt to avoid detection by IDPSs.

Most IDPSs use multiple detection methodologies, either separately or integrated, to provide more broad and accurate detection. The primary classes of detection methodologies are as follows:

- **Signature-based**, which compares known threat signatures to observed events to identify incidents. This is very effective at detecting known threats but largely ineffective at detecting unknown threats and many variants on known threats. Signature-based detection cannot track and understand the state of complex communications, so it cannot detect most attacks that comprise multiple events.
- **Anomaly-based detection**, which compares definitions of what activity is considered normal against observed events to identify significant deviations. This method uses profiles that are developed by monitoring the characteristics of typical activity over a period of time. The IDPS then compares the characteristics of current activity to thresholds related to the profile. Anomaly-based detection methods can be very effective at detecting previously unknown threats. Common problems with anomaly-based detection are inadvertently including malicious activity within a profile, establishing profiles that are not sufficiently complex to reflect real-world computing activity, and generating many false positives.
- **Stateful protocol analysis**, which compares 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. It is capable of understanding and tracking the state of protocols that have a notion of state, which allows it to detect many attacks that other methods cannot. Problems with stateful protocol analysis include that it is often very difficult or impossible to develop completely accurate models of protocols, it is very resource-intensive, and it cannot detect attacks that do not violate the characteristics of generally acceptable protocol behavior.

3. IDPS Technologies

This section provides an overview of IDPS technologies. The information presented in this section applies to all types of IDPS products; additional information specific to each product type is presented in Sections 4 through 7. This section first covers the major components of IDPS technologies and explains the architectures typically used for deploying the components. It also provides a high-level description of the security capabilities of the technologies, including the methodologies they use to identify suspicious activity. The rest of the section discusses the management capabilities of the technologies, including detailed recommendations for implementation and operation.

3.1 Components and Architecture

This section describes the major components of IDPS solutions and illustrates the most common network architectures for these components.

3.1.1 Typical Components

The typical components in an IDPS solution are as follows:

- **Sensor or Agent.** Sensors and agents monitor and analyze activity. The term *sensor* is typically used for IDPSs that monitor networks, including network-based, wireless, and network behavior analysis technologies. The term *agent* is typically used for host-based IDPS technologies.
- **Management Server.** A *management server* is a centralized device that receives information from the sensors or agents and manages them.[7] Some management servers perform analysis on the event information that the sensors or agents provide and can identify events 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*. Management servers are available as both appliance and software-only products. Some small IDPS deployments do not use any management servers, but most IDPS deployments do. In larger IDPS deployments, there are often multiple management servers, and in some cases there are two tiers of management servers.
- **Database Server.** A *database server* is a repository for event information recorded by sensors, agents, and/or management servers. Many IDPSs provide support for 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.

3.1.2 Network Architectures

IDPS components can be connected to each other through an organization's standard networks or through a separate network strictly 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. Also, each sensor or agent host is unable to pass any traffic between its management interface and any of its other network interfaces. The

[7] Because this publication focuses on enterprise IDPS deployment, it assumes that management servers are used with sensors and agents. However, some types of IDPS sensors and agents can be deployed standalone, and managed and monitored directly by administrators without using a management server.

management servers, database servers, and consoles are attached to the management network only. This architecture effectively isolates the management network from the production networks. The benefits of doing this are to conceal the existence and identity of the IDPS from attackers; to protect the IDPS from attack; and to ensure that the IDPS has adequate bandwidth to function under adverse conditions (e.g., worm attack or distributed denial of service [DDoS] on the monitored networks). Disadvantages of using a management network include the additional costs in networking equipment and other hardware (e.g., PCs for the consoles) and the inconvenience for IDPS users and administrators of using separate computers for IDPS management and monitoring.

If an IDPS is deployed without a separate management network, another 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. For example, misconfiguration of the VLAN could lead to the exposure of IDPS data. Another concern is that under adverse conditions, such as DDoS attacks or major malware incidents, the network devices shared by the organization's primary networks and VLAN might become completely saturated, negatively impacting the availability and performance of the IDPS.

3.2 Security Capabilities

Most IDPS technologies can provide a wide variety of security capabilities. Sections 3.2.1 through 3.2.4 describe common security capabilities, divided into four categories: information gathering, logging, detection, and prevention, respectively.

3.2.1 Information Gathering Capabilities

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.

3.2.2 Logging Capabilities

IDPSs typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDPS and other logging sources. Data fields commonly used by IDPSs include event date and time, event type, importance rating (e.g., priority, severity, impact, confidence), and prevention action performed (if any). Specific types of IDPSs log additional data fields, such as network-based IDPSs performing packet captures and host-based IDPSs recording user IDs. IDPS technologies typically permit administrators to store logs locally and send copies of logs to centralized logging servers (e.g., syslog, security information and event management software). Generally, logs should be stored both locally and centrally to support the integrity and availability of the data (e.g., a compromise of the IDPS could allow attackers to alter or destroy its logs).[8] Also, IDPSs should have their clocks synchronized using the Network Time Protocol (NTP) or through frequent manual adjustments so that their log entries have accurate timestamps.[9]

[8] For additional information on log management, see NIST SP 800-92, *Guide to Computer Security Log Management*, which is available at http://csrc.nist.gov/publications/nistpubs/.

[9] NIST SP 800-86, *Guide to Integrating Forensic Techniques into Incident Response*, contains additional information on the importance of clock synchronization for investigating events and correlating information across systems. The publication is available at http://csrc.nist.gov/publications/nistpubs/.

3.2.3 Detection 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, such as setting the prevention actions to be performed for particular alerts. Technologies vary widely in their tuning and customization capabilities. Typically, the more powerful a product's tuning and customization capabilities are, the more its detection accuracy can be improved from the default configuration. Organizations should carefully consider the tuning and customization capabilities of IDPS technologies when evaluating products. Examples of such 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 x failed connection attempts in 60 seconds, or x characters for a filename length. Thresholds are most often used for anomaly-based detection and stateful protocol analysis.
- **Blacklists and Whitelists.** A *blacklist* is a list of discrete entities, such as hosts, TCP or UDP port numbers, ICMP types and codes, applications, usernames, URLs, filenames, or file extensions, that have been previously determined to be associated with malicious activity. Blacklists, also known as *hot lists*, are typically used to allow IDPSs to recognize and block activity that is highly likely to be malicious, and may also be used to assign a higher priority to alerts that match entries on the blacklists. Some IDPSs generate dynamic blacklists that are used to temporarily block recently detected threats (e.g., activity from an attacker's IP address). A *whitelist* is a list of discrete entities that are known to be benign. Whitelists are typically used on a granular basis, such as protocol-by-protocol, to reduce or ignore false positives involving known benign activity from trusted hosts. Whitelists and blacklists are most commonly used in signature-based detection and stateful protocol analysis.
- **Alert Settings.** Most IDPS technologies allow administrators to customize each alert type. Examples of actions that can be performed on an alert type include the following:
  - Toggling it on or off[10]
  - Setting a default priority or severity level
  - Specifying what information should be recorded and what notification methods (e.g., e-mail, pager) should be used
  - Specifying which prevention capabilities should be used.

  Some products also suppress alerts if an attacker generates many alerts in a short period of time, and may also temporarily ignore all future traffic from the attacker. This is to prevent the IDPS from being overwhelmed by alerts.
- **Code Viewing and Editing.** Some IDPS technologies permit administrators to see some or all of the detection-related code. This is usually limited to signatures, but some technologies allow administrators to see additional code, such as programs used to perform stateful protocol analysis.

[10] In some IDPS technologies, turning off an alert also disables related detection capabilities; in other products, the detection processing is still done but an alert message is not generated. For technologies in the first category, shutting off unneeded alerts can reduce the load on the IDPS.

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 all detection-related code and write new code (e.g., new signatures) is necessary to fully customize certain types of detection capabilities. For example, a particular alert might be generated by a complex series of events involving several code modules; customizing the IDPS to understand organization-specific characteristics might not be possible without editing the code directly. Editing the code requires programming and intrusion detection skills; also, some IDPSs use proprietary programming languages, which would necessitate the programmer learning a new language. Bugs introduced into the code during the customization process could cause the IDPS to function incorrectly or fail altogether, so administrators should treat code customization as they would any other alteration of production systems' code.

Administrators should review tuning and customizations periodically to ensure that they are still accurate. For example, whitelists and blacklists should be checked regularly and all entries validated to ensure that they are still accurate and necessary. Thresholds and alert settings might need to be adjusted periodically to compensate for changes in the environment and in threats. Edits to detection code might need to be replicated whenever the product is updated (e.g., patched, upgraded). Administrators should also ensure that any products collecting baselines for anomaly-based detection have their baselines rebuilt periodically as needed to support accurate detection.

3.2.4 Prevention 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 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.

3.3 Management

Most IDPS products offer similar management capabilities. This section discusses major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently. It also briefly discusses the skills needed for IDPS management and provides recommendations for gaining these skills.

3.3.1 Implementation

Once an IDPS product has been selected, the administrators need to design an architecture, perform IDPS component testing, and deploy and secure the IDPS components. Sections 3.3.1.1 through 3.3.1.3 provide more information on these actions.

3.3.1.1 Architecture Design

The first step in IDPS implementation is designing an architecture. Architectural considerations include the following:

- Where the sensors or agents should be placed

- How reliable the solution should be and what measures should be used to achieve that reliability, such as having multiple sensors monitor the same activity in case a sensor fails, or using multiple management servers so that a backup server can be used in case the primary server fails
- Where the other components of the IDPS will be located (e.g., management servers, database servers, consoles), and how many of each component are needed to achieve the necessary usability, redundancy, and load balancing goals
- With which other systems the IDPS needs to interface, including the following:
  - Systems to which it provides data, such as security information and event management software, centralized log servers, e-mail servers, and paging systems
  - Systems on which it initiates prevention responses (e.g., firewalls, routers, switches)
  - Systems that manage IDPS components, such as network management software (for a management network) or patch management software (for keeping consoles' operating systems and applications fully up-to-date)
- Whether or not a management network will be used; if so, what its design will be, and if not, how the IDPS communications will be protected on the standard networks
- What other security controls and technologies need to be altered to accommodate IDPS deployment, such as changing firewall rulesets to allow IDPS components to communicate.

3.3.1.2 Component Testing and Deployment

Organizations should consider implementing the components in a test environment first, instead of a production environment, to reduce the likelihood of implementation problems disrupting the production networks. When the components are being deployed to production networks, organizations should initially activate only a few IDPS sensors or agents, with their prevention capabilities disabled. Because a new deployment is likely to generate a large number of false positives until fully tuned and customized, activating many sensors or agents at once might overwhelm the management servers and consoles, making it difficult for administrators to perform tuning and customization. Many false positives are likely to be the same across sensors or agents, so it is helpful to identify such false positives either during the testing process or when deploying the first few sensors or agents, so that those false positives can be addressed before widespread deployment. A phased deployment of sensors or agents is also helpful in identifying potential problems with scalability.

Implementing an IDPS can necessitate brief network or system outages for component installation. As mentioned above, performing a deployment in a test environment can be very helpful in identifying likely implementation problems, so that they can be mitigated appropriately when the production deployment occurs.

Appliance-based IDPS components are typically simple to deploy. Administrators might need to perform software updates or signature updates to ensure the IDPS software is current. Otherwise, administrators usually just need to provide power and connect network cables, boot the appliance, and perform some basic configuration (e.g., enter a product license key, assign a name to the sensor).

Software-based IDPS components usually take more time to deploy than appliance-based components. The organization first needs to acquire the appropriate hardware, which might include purchasing high-bandwidth network cards and otherwise ensuring that the hardware is robust enough for the IDPS. Next, administrators need to install an operating system (OS) that is compatible with the IDPS software, and

then harden the host as much as possible. Hardening should include updating the OS, services, and applications, including the IDPS software. Administrators also need to perform basic configuration of the IDPS software, just as is done for appliance-based IDPS components.

After deploying either appliance-based or software-based IDPS components, considerable effort may be needed to configure the products' detection and prevention capabilities, depending on the type of IDPS being deployed. Without performing this configuration work, some IDPSs might be capable of detecting only a small number of older, easily identified attacks.

3.3.1.3 Securing IDPS Components

Securing IDPS components is very important because IDPSs are often targeted by attackers. If an attacker can compromise an IDPS, it can be rendered useless in detecting subsequent attacks against other hosts. Also, IDPSs often contain sensitive information such as host configurations and known vulnerabilities that could be helpful in planning additional attacks. In addition to hardening software-based IDPS components and ensuring that all IDPS components are fully up-to-date, administrators should perform additional actions to ensure that the IDPS components themselves are secured appropriately. Specific security recommendations are as follows:

- Administrators should create separate accounts for each user and administrator of the IDPS, and assign each account only the necessary privileges.
- Administrators should configure firewalls, routers, and other packet filtering devices to limit direct access to all IDPS components to only those hosts that need such access.
- Administrators should ensure that all IDPS management communications are protected appropriately, either through physical (e.g., management network) or logical (e.g., management VLAN) separation, or through encryption of communications. If encryption is used for protection, it should be performed using FIPS-approved encryption algorithms.[11] Many products encrypt their communications using Transport Layer Security (TLS); for products that do not provide sufficient protection through encryption, organizations should consider using a virtual private network (VPN) or other encrypted tunneling method to protect the traffic.

Some organizations also require the use of strong authentication for remote access to IDPS components, such as two-factor authentication. This provides an additional layer of security.

3.3.2 Operation and Maintenance

Nearly all IDPS products are designed to be operated and maintained through a graphical user interface (GUI), also known as the *console*. The console typically permits administrators to configure and update the sensors and management servers, as well as monitor their status (e.g., agent failure, packet dropping). Administrators can also manage user accounts, customize reports, and perform many other functions using the console. IDPS users can also perform many functions through the console, including monitoring and analyzing the IDPS data and generating reports. Most IDPSs permit administrators to set up individual user accounts for each administrator and user, and to grant each account only the privileges necessary for each person's role. The console often reflects this by showing different menus and options based on the currently authenticated account's designated role. Some products also provide finer-grained

[11] Federal agencies must use Federal Information Processing Standard (FIPS) approved encryption algorithms contained in validated cryptographic modules. The Cryptographic Module Validation Program (CMVP) at NIST coordinates FIPS testing; the CMVP Web site is located at http://csrc.nist.gov/cryptval/. See http://csrc.nist.gov/cryptval/des.htm for information on FIPS-approved symmetric key algorithms. FIPS 140-2, *Security Requirements for Cryptographic Modules*, is available at http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

access control, such as specifying for which sensors or agents particular users can monitor or analyze data or generate reports or particular administrators can alter configurations. This allows a large IDPS deployment to be divided into logical units for operational purposes.

Some IDPS products also offer command-line interfaces (CLI). Unlike GUI consoles, which are typically used for remote management of sensors or agents and management servers, CLIs are typically used for local management of those components. Sometimes a CLI can be reached remotely through an encrypted connection established through secure shell (SSH) or other means. Consoles are typically much easier to use than CLIs, and CLIs often provide only some of the functionality that consoles provide.

The rest of this section provides additional information on the operation and maintenance of IDPSs. Section 3.3.2.1 describes how users can make effective use of consoles in their daily IDPS tasks. Section 3.3.2.2 provides more information on ongoing maintenance of the technologies, with Section 3.3.2.3 focusing specifically on acquiring and applying updates.

3.3.2.1 Typical Use

Most IDPS consoles offer many features to assist users in their daily tasks. For example, most consoles offer drill-down capabilities, which means that when a user examines an alert, more details and information are available in layers.[12] This allows users to see basic information on many alerts at once, and to show additional information on particular events of interest as needed. Some products allow users to see extensive supporting information, such as packet captures (both raw and parsed with a protocol analyzer) and related alerts (e.g., other alerts for the same source or destination), as well as documentation on the alert itself. Generally, the more data that the IDPS records, the easier it is for analysts to determine what has happened. Some consoles also offer incident response features, such as turning an alert into an incident case and providing workflow mechanisms that allow users to document additional information on the alert and route the alert to particular users or groups of users for further review.

Most consoles also offer various reporting functions. For example, administrators or users might be able to use the console to have certain reports run at set times and to e-mail or transfer the reports to the appropriate users or hosts. Many consoles also allow users to generate reports as needed (including reports for specific incidents) and to customize reports as needed. If an IDPS product stores its logs in a database or in an easily parsed file format (e.g., comma-separated values in a text file), database queries or scripts can also be used to generate custom reports, particularly if the console does not offer sufficiently flexible report customization.

3.3.2.2 Ongoing Solution Maintenance

Administrators should maintain IDPSs on an ongoing basis. This should include the following:

- Monitoring the IDPS components themselves for operational and security issues
- Periodically verifying that the IDPS is functioning properly (e.g., processing events, alerting appropriately on suspicious activity)[13]

---

[12] A detailed discussion of the analysis of IDPS data is outside the scope of this document. For more information, see NIST SP 800-86, *Guide to Integrating Forensic Techniques into Incident Response*, which is available at http://csrc.nist.gov/publications/nistpubs/. NIST SP 800-86 discusses the analysis of data from IDPSs and other sources of security event information.

[13] One way of verifying component functionality is to perform periodic testing of the IDPS, either in a test environment or a production environment. Performing such testing in a production environment can inadvertently disrupt operations. See Section 9 for additional information on IDPS testing.

- Performing regular vulnerability assessments
- Receiving notifications from vendors of security problems with IDPS components (including OSs and non-IDPS applications) and responding appropriately to those notifications
- Receiving notifications from the IDPS vendor of updates, and performing testing and deployment of the updates. Updates are described in Section 3.3.2.3.

3.3.2.3 Acquiring and Applying Updates

There are two types of IDPS updates: software updates and signature updates. *Software updates* fix bugs in the IDPS software or add new functionality, while *signature updates* add new detection capabilities or refine existing detection capabilities (e.g., reducing false positives). For many IDPSs, signature updates cause program code to be altered or replaced, so they are really a specialized form of software update. For other IDPSs, signatures are not written in code, so a signature update is a change to the configuration data for the IDPS.

Software updates can include any or all IDPS components, including sensors, agents, management servers, and consoles. Software updates for sensors and management servers, particularly appliance-based devices, are often applied by replacing an existing IDPS CD with a new one and rebooting the device. Many IDPSs run the software directly from the CD, so that no software installation is required. Other components, such as agents, require an administrator to install software or apply patches, either manually on each host or automatically through IDPS management software. Some vendors make software and signature updates available for download from their Web sites or other servers; often, the administrator interfaces for IDPSs have features for downloading and installing such updates.

Administrators should verify the integrity of updates before applying them, because updates could have been inadvertently or intentionally altered or replaced. The recommended verification method depends on the update's format, as follows:

- **Files downloaded from a Web site or FTP site.** Administrators should compare file checksums provided by the vendor with checksums that they compute for the downloaded files.
- **Update downloaded automatically through the IDPS user interface.** If an update is downloaded as a single file or a set of files, either checksums provided by the vendor should be compared to checksums generated by the administrator, or the IDPS user interface itself should perform some sort of integrity check. In some cases, updates might be downloaded and installed as one action, precluding checksum verification; the IDPS user interface should check each update's integrity as part of this.
- **Removable media** (e.g., CD, DVD). Vendors may not provide a specific method for customers to verify the legitimacy of removable media apparently sent by the vendors. If media verification is a concern, administrators should contact their vendors to determine how the media can be verified, such as comparing vendor-provided checksums to checksums computed for files on the media, or verifying digital signatures on the media's contents to ensure they are valid. Administrators should also consider scanning the media for malware, with the caveat that false positives might be triggered by IDPS signatures for malware on the media.

IDPSs are typically designed so that applying software and signature updates has no effect on existing tuning and customization settings. The primary exception is code customization, which often has to be repeated when code updates from the vendor are installed. For any IDPS, administrators should back up configuration settings periodically and before applying software or signature updates to ensure that existing settings are not inadvertently lost.

Administrators should test software and signature updates before applying them except for emergency situations (e.g., a signature identifies a new, active threat that is damaging the organization and cannot otherwise be detected or blocked). It is beneficial to have at least one sensor or agent host (one for each type of agent) that is used strictly for testing updates. New detection capabilities can often cause large numbers of alerts to be triggered, so testing signature updates on a single sensor or agent host, even briefly (e.g., loading the update and observing how the IDPS functions when monitoring typical activity), can help to identify signatures that are likely to be problematic and should possibly be disabled. In non-emergency situations, software and signature updates should be tested and deployed using the same practices that would be used for updating any other major security controls, such as firewalls and antivirus software. When updates are deployed into production, administrators should be ready to disable particular signatures or perform other minor reconfigurations as needed.

3.3.3 Building and Maintaining Skills

Various skills are needed for IDPS implementation, operation, and maintenance, including the following:

- The administrators implementing IDPS components need to understand the basics of system administration, network administration, and information security.
- The administrators tuning and customizing the IDPS need reasonably comprehensive knowledge of information security and IDPS principles. An understanding of incident response principles and the organization's incident response policies and procedures is also recommended. Administrators should also have an understanding of the network protocols, applications, and operating systems to be monitored by the IDPS.
- Programming skills might also be needed for extensive code customization, report writing, and other tasks.

Skills in IDPS principles can be built and maintained through many methods, including training, technical conferences, books and other technical references, and mentoring programs. Knowledge of specific IDPS products can also be gained through several methods, including the following:

- **Vendor Training.** Many vendors of IDPS products offer one or more training courses for people that will be administering or using their products. Training courses are often hands-on, permitting attendees to learn how to use the technology in a non-production environment.
- **Product Documentation.** Most products offer several manuals, such as an installation guide, a user's guide, and an administrator's guide. Some also offer separate guides or databases that provide supplemental information for alerts and signatures.
- **Technical Support.** Most vendors offer technical support for their customers, either as part of purchasing a product or for an additional fee. Support is used primarily to resolve problems and clarify the capabilities of the product to its users and administrators.
- **Professional Services.** Some vendors offer professional services, which is essentially consulting services provided by the vendor. For example, an organization could pay a vendor to write custom signatures or reports, or to assist administrators in understanding how to tune and customize their sensors effectively.
- **User Communities.** Some products have active user communities, which typically function through mailing lists or online forums. Users can exchange information and code with each other, and assist each other in troubleshooting problems. Although user communities can be a source of information, administrators and users should be cautious when using them, because posting details about an

organization's IDPS configuration or problems could inadvertently reveal sensitive information about the organization's security infrastructure, systems, and networks.

3.4 Summary

The typical components in an IDPS solution are sensors or agents, management servers, database servers, and consoles. Sensors and agents monitor and analyze activity; sensors are used to monitor networks and agents to monitor hosts. Management servers receive information from sensors or agents and manage them. Database servers are repositories for event information recorded by the sensors or agents and management servers. Consoles are programs that provide interfaces for IDPS users and administrators. These components can be connected to each other through an organization's standard networks or through a separate network strictly designed for security software management known as a management network. A management network helps to protect the IDPS from attack and to ensure it has adequate bandwidth under adverse conditions. A virtual management network can be created using a virtual local area network (VLAN); this provides protection for IDPS communications, but not as much protection as a management network would provide.

Most IDPSs can provide a wide variety of security capabilities. Some products offer information gathering capabilities, such as collecting information on hosts or networks from observed activity. IDPSs also typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDPS and other logging sources. Generally, logs should be stored both locally and centrally to support the integrity and availability of the data.

IDPSs typically offer extensive, broad detection capabilities. 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. Typically, the more powerful a product's tuning and customization capabilities are, the more its detection accuracy can be improved from the default configuration. Examples of these capabilities are thresholds, blacklists and whitelists, alert settings, and code editing. Organizations should carefully consider the tuning and customization capabilities of IDPSs when evaluating products. Administrators should review tuning and customizations periodically to ensure that they are still accurate. Administrators should also ensure that any products collecting baselines for anomaly-based detection have those baselines rebuilt periodically as needed to support accurate detection.

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 includes enabling or disabling prevention, as well as specifying which type of prevention capability should be used.

Once an IDPS product has been selected, the administrators need to design an architecture, perform IDPS component testing, and deploy and secure the IDPS components. There are many architectural considerations, including component placement, solution reliability, interoperability with other systems, management network architecture, and necessary changes to other security controls. Before performing a production implementation, organizations should consider implementing the components in a test environment first to reduce the likelihood of implementation problems disrupting production. When the components are being deployed to production networks, organizations should initially activate only a few IDPS sensors or agents. Because a new deployment is likely to generate a large number of false positives until fully tuned and customized, activating many sensors or agents at once might overwhelm the management servers and consoles, making it difficult for administrators to perform tuning and customization.

In addition to hardening software-based IDPS components and ensuring that all IDPS components are fully up-to-date, administrators should perform additional actions to ensure that the IDPS components themselves are secured appropriately. Examples include creating separate accounts for each IDPS user and administrator, restricting network access to IDPS components, and ensuring that IDPS management communications are protected appropriately. All encryption used for protection should be performed using FIPS-approved encryption algorithms.

Administrators should maintain IDPSs on an ongoing basis. This should include monitoring the IDPS components for operational and security issues, performing regular vulnerability assessments, responding appropriately to vulnerabilities in the IDPS components, and testing and deploying IDPS software and signature updates. Administrators should verify the integrity of updates before applying them, because updates could have been inadvertently or intentionally altered or replaced. Administrators should test software and signature updates before applying them, except for emergency situations. Administrators should also back up configuration settings periodically and before applying software or signature updates to ensure that existing settings are not inadvertently lost.

This page has been left blank intentionally.

4. Network-Based IDPS

A network-based IDPS monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity. This section provides a detailed discussion of network-based IDPS technologies. First, it contains a brief overview of TCP/IP, which is background material for understanding the rest of Section 4. Next, it covers the major components of network-based IDPSs and explains the architectures typically used for deploying the components. It also examines the security capabilities of the technologies in depth, including the methodologies they use to identify suspicious activity. The rest of the section discusses the management capabilities of the technologies and provides recommendations for implementation and operation.

4.1 Networking Overview

TCP/IP is widely used throughout the world to provide network communications. TCP/IP communications are composed of four layers that work together. When a user wants to transfer data across networks, the data is passed from the highest layer through intermediate layers to the lowest layer, with each layer adding more information. The lowest layer sends the accumulated data through the physical network; the data is then passed up through the layers to its destination. Essentially, the data produced by a layer is encapsulated in a larger container by the layer below it. The four TCP/IP layers, from highest to lowest, are shown in Figure 4-1.

Application Layer. This layer sends and receives data for particular applications, such as Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), and Simple Mail Transfer Protocol (SMTP).
Transport Layer. This layer provides connection-oriented or connectionless services for transporting application layer services between networks. The transport layer can optionally ensure the reliability of communications. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are commonly used transport layer protocols.
Internet Protocol (IP) Layer (also known as Network Layer). This layer routes packets across networks. IPv4 is the fundamental network layer protocol for TCP/IP. Other commonly used protocols at the network layer are IPv6, Internet Control Message Protocol (ICMP), and Internet Group Management Protocol (IGMP).
Hardware Layer (also known as Data Link Layer). This layer handles communications on the physical network components. The best known data link layer protocol is Ethernet.

Figure 4-1. TCP/IP Layers

The four TCP/IP layers work together to transfer data between hosts. Network-based IDPSs typically perform most of their analysis at the application layer. They also analyze activity at the transport and network layers both to identify attacks at those layers and to facilitate the analysis of the application layer activity (e.g., a TCP port number may indicate which application is being used). Some network-based IDPSs also perform limited analysis at the hardware layer. Sections 4.1.1 through 4.1.4 describe each layer in greater detail.

4.1.1 Application Layer

The application layer enables applications to transfer data between an application server and client. An example of an application layer protocol is Hypertext Transfer Protocol (HTTP), which transfers data between a Web server and a Web browser. Other common application layer protocols include Domain Name System (DNS), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Simple

Network Management Protocol (SNMP). There are hundreds of unique application layer protocols in common use, and many more that are not so common. Regardless of the protocol in use, application data is generated and then passed to the transport layer for further processing.

4.1.2 Transport Layer

The transport layer is responsible for packaging data so that it can be transmitted between hosts. Most applications that communicate over networks rely on the transport layer to ensure reliable delivery of data. Generally, this is accomplished by using TCP. When loss of some application data is not a concern (e.g., streaming audio, video), or the application itself ensures reliable delivery of data, UDP is typically used. UDP is connectionless; one host simply sends data to another host without any preliminary negotiations. Each TCP or UDP packet has a source port number and a destination port number. One of the ports is associated with a server application on one system; the other port is associated with a corresponding client application on the other system. Client systems typically select any available port number for application use, whereas server systems usually have a static port number dedicated to each application. Although UDP and TCP ports are very similar, they are distinct from each other and are not interchangeable.

4.1.3 Network Layer

The network layer, also known as the IP layer, is responsible for handling the addressing and routing of data that it receives from the transport layer. After the network layer has encapsulated the transport layer data, the resulting logical units are referred to as *packets*. Each packet contains a *header*, which is composed of various *fields* that specify characteristics of the transport protocol in use; optionally, packets may also contain a *payload*, which holds the application data. The IP header contains a field called IP Version, which indicates which version of IP is in use. Typically this is set to 4 for IPv4; but the use of IPv6 is increasing, so this field may be set to 6 instead.[14] Other significant IP header fields are as follows:

- **Source and Destination IP Addresses.** These are the "from" and "to" addresses that are intended to indicate the endpoints of the communication.[15] Examples of IP addresses are 10.3.1.70 (IPv4) and 1000::2F:8A:400:427:9BD1 (IPv6).
- **IP Protocol Number.** This indicates which network or transport layer protocol the IP payload contains.[16] Commonly used IP numbers include 1 (ICMP), 6 (TCP), 17 (UDP), and 50 (Encapsulating Security Payload [ESP]).

The network layer is also responsible for providing error and status information involving the addressing and routing of data; it does this with ICMP. ICMP is a connectionless protocol that makes no attempt to guarantee that its error and status messages are delivered. Because it is designed to transfer limited information, not application data, ICMP does not have ports; instead, it has message types, which indicate the purpose of each ICMP message.[17] Some message types also have message codes, which can be thought of as subtypes. For example, the ICMP message type Destination Unreachable has several

[14] There are other possible IP version numbers as well, although none are commonly used. The official list of valid IP Version field values is available at http://www.iana.org/assignments/version-numbers. This document assumes the use of IPv4, unless otherwise specified.

[15] IP addresses are often inaccurate or misleading for identifying the actual endpoints of communication.

[16] The official list of valid IP Protocol Number values is available at http://www.iana.org/assignments/protocol-numbers.

[17] The current list of valid ICMP types is available at http://www.iana.org/assignments/icmp-parameters.

possible message codes that indicate what is unreachable (e.g., network, host, protocol). Most ICMP message types are not intended to elicit a response.[18]

4.1.4 Hardware Layer

As the name implies, the hardware layer, also called the data link layer, involves the physical components of the network, including cables, routers, switches, and network interface cards (NIC). The hardware layer also includes various hardware layer protocols, with Ethernet being the most widely used. Ethernet relies on the concept of a media access control (MAC) address, which is a unique six-byte value (such as 00-02-B4-DA-92-2C) that is permanently assigned to a particular NIC.[19] Each *frame*, the logical unit at the hardware layer, contains two MAC addresses, which indicate the MAC address of the NIC that just routed the frame and the MAC address of the next NIC to which the frame is being sent. As a frame passes through networking equipment (such as routers and firewalls) on its way between the original source host and the final destination host, the MAC addresses are updated to refer to the local source and destination. Several separate hardware layer transmissions may be linked together within a single network layer transmission.

In addition to the MAC addresses, each frame also contains an EtherType value, which indicates the protocol that the frame's payload contains (typically IP or Address Resolution Protocol [ARP]).[20] When IP is used, each IP address maps to a particular MAC address. (Because multiple IP addresses can map to a single MAC address, a MAC address does not necessarily uniquely identify an IP address.)

4.2 Components and Architecture

This section describes the major components of typical network-based IDPSs and illustrates the most common network architectures for these components. It also provides recommendations for the placement of network-based IDPS sensors.

4.2.1 Typical Components

A typical network-based IDPS is composed of sensors, one or more management servers, multiple consoles, and optionally one or more database servers (if the network-based IDPS supports their use). All of these components are similar to other types of IDPS technologies, except for the sensors. A network-based IDPS sensor monitors and analyzes network activity on one or more network segments. The network interface cards that will be performing monitoring are placed into *promiscuous mode*, which means that they will accept all incoming packets that they see, regardless of their intended destinations. Most IDPS deployments use multiple sensors, with large deployments having hundreds of sensors. Sensors are available in two formats:

- **Appliance.** An appliance-based sensor is comprised of specialized hardware and sensor software. The hardware is typically optimized for sensor use, including specialized NICs and NIC drivers for efficient capture of packets, and specialized processors or other hardware components that assist in analysis. Parts or all of the IDPS software might reside in firmware for increased efficiency.

---

[18] ICMP is designed to limit responses, particularly to error messages. If ICMP had not been designed in this way, message loops could occur. For example, if Host A received an ICMP error message from Host B and responded with an error message, and Host B responded to that error message with an error message, the two hosts could continue sending error messages regarding the error messages.

[19] Various software utilities are publicly available that allow people to configure systems to spoof other MAC addresses. There have also been cases in which manufacturers accidentally created NICs with duplicate MAC addresses.

[20] EtherType value 0x0800 is IP, while 0x0806 is ARP. See http://www.iana.org/assignments/ethernet-numbers for more information on EtherType values.

Appliances often use a customized, hardened operating system (OS) that administrators are not intended to access directly.

- **Software Only.** Some vendors sell sensor software without an appliance. Administrators can install the software onto hosts that meet certain specifications. The sensor software might include a customized OS, or it might be installed onto a standard OS just as any other application would.

4.2.2 Network Architectures and Sensor Locations

Organizations should consider using management networks for their network-based IDPS deployments whenever feasible. If an IDPS is deployed without a separate management network, organizations should consider whether or not a VLAN is needed to protect the IDPS communications.

In addition to choosing the appropriate network for the components, administrators also need to decide where the IDPS sensors should be located. Sensors can be deployed in one of two modes:

- **Inline.** An *inline sensor* is deployed so that the network traffic it is monitoring must pass through it, much like the traffic flow associated with a firewall. In fact, some inline sensors are hybrid firewall/IDPS devices, while others are simply IDPSs. The primary motivation for deploying IDPS sensors inline is to enable them to stop attacks by blocking network traffic. Inline sensors are typically placed where network firewalls and other network security devices would be placed—at the divisions between networks, such as connections with external networks and borders between different internal networks that should be segregated. Inline sensors that are not hybrid firewall/IDPS devices are often deployed on the more secure side of a network division so that they have less traffic to process. Figure 4-2 shows such a deployment. Sensors can also be placed on the less secure side of a network division to provide protection for and reduce the load on the dividing device, such as a firewall.

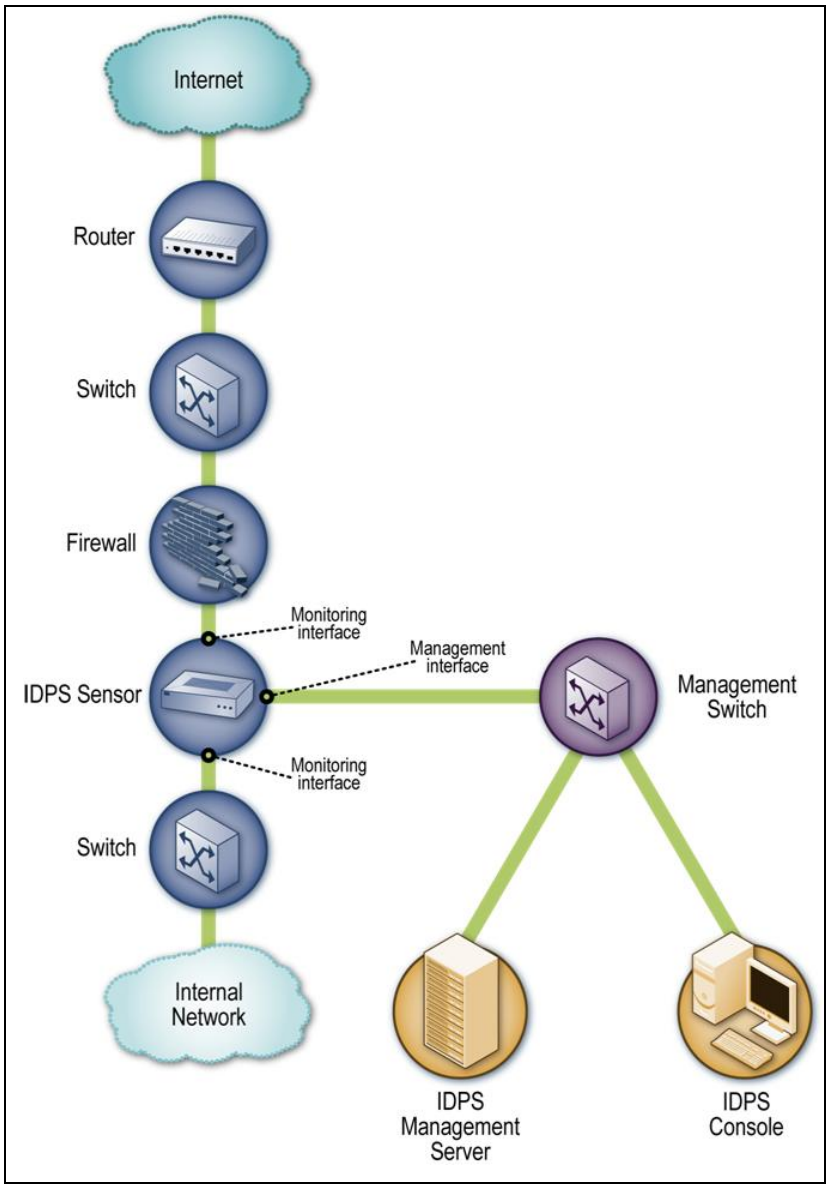

Figure 4-2. Inline Network-Based IDPS Sensor Architecture Example

- **Passive.** A *passive sensor* is deployed so that it monitors a copy of the actual network traffic; no traffic actually passes through the sensor. Passive sensors are typically deployed so that they can monitor key network locations, such as the divisions between networks, and key network segments, such as activity on a demilitarized zone (DMZ) subnet. Passive sensors can monitor traffic through various methods, including the following:
  - **Spanning Port.** Many switches have a *spanning port*, which is a port that can see all network traffic going through the switch. Connecting a sensor to a spanning port can allow it to monitor traffic going to and from many hosts. Although this monitoring method is relatively easy and inexpensive, it can also be problematic. If a switch is configured or reconfigured incorrectly, the spanning port might not be able to see all the traffic. Another problem with spanning ports is that their use can be resource-intensive; when a switch is under heavy loads, its spanning port might

not be able to see all traffic, or spanning might be temporarily disabled. Also, many switches have only one spanning port, and there is often a need to have multiple technologies, such as network monitoring tools, network forensic analysis tools, and other IDPS sensors, monitor the same traffic.

- **Network Tap.** A *network tap* is a direct connection between a sensor and the physical network media itself, such as a fiber optic cable. The tap provides the sensor with a copy of all network traffic being carried by the media. Installing a tap generally involves some network downtime, and problems with a tap could cause additional downtime. Also, unlike spanning ports, which are usually already present throughout an organization, network taps need to be purchased as add-ons to the network.

- **IDS Load Balancer.** An *IDS load balancer* is a device that aggregates and directs network traffic to monitoring systems, including IDPS sensors. A load balancer can receive copies of network traffic from one or more spanning ports or network taps and aggregate traffic from different networks (e.g., reassemble a session that was split between two networks). The load balancer then distributes copies of the traffic to one or more listening devices, including IDPS sensors, based on a set of rules configured by an administrator. The rules tell the load balancer which types of traffic to provide to each listening device. Common configurations include the following:

  - **Send all traffic to multiple IDPS sensors.** This could be done for high availability or to have multiple types of IDPS sensors perform concurrent analysis of the same activity.

  - **Dynamically split the traffic among multiple IDPS sensors based on volume.** This is typically done to perform load balancing so that no sensor is overwhelmed with the amount of traffic and corresponding analysis.

  - **Split the traffic among multiple IDPS sensors based on IP addresses, protocols, or other characteristics.** This could be done for load balancing purposes, such as having one IDPS sensor dedicated to Web activity and another IDPS sensor monitoring all other activity. Splitting traffic could also be done to perform more detailed analysis of certain types of traffic (e.g., activity involving the most important hosts).

  Splitting traffic among multiple IDPS sensors can cause a reduction in detection accuracy if related events or portions of a single event are seen by different sensors. For example, suppose that two sensors each see different steps of an attack; if each step is considered benign on its own but the two steps in sequence are malicious, then the attack might not be recognized.

Figure 4-3 shows examples of passive sensors connected to the monitored network using IDS load balancers, network taps, and spanning ports.

As explained in Section 4.3.4, most techniques for having a sensor prevent intrusions require that the sensor be deployed in inline mode, not passive. Because passive techniques monitor a copy of the traffic, they typically provide no reliable way for a sensor to stop the traffic from reaching its destination. 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. Generally, organizations should deploy sensors inline if prevention methods will be used and passive if they will not.

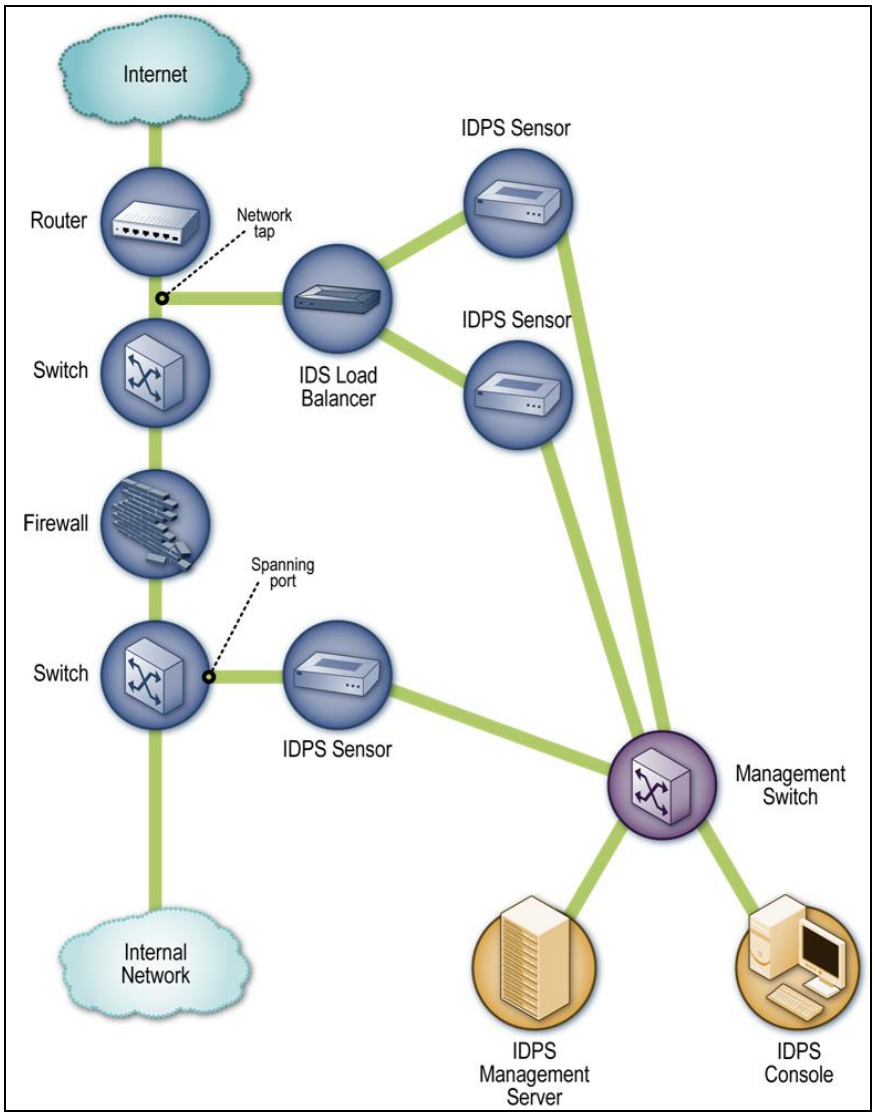

Figure 4-3. Passive Network-Based IDPS Sensor Architecture Example

4.3 Security Capabilities

Network-based IDPS products provide a wide variety of security capabilities. Sections 4.3.1 through 4.3.4 describe common security capabilities, divided into four categories: information gathering, logging, detection, and prevention, respectively. Some network-based IDPS products also provide some security information and event management (SIEM) capabilities; see Section 8.2.2 for information on SIEM.

4.3.1 Information Gathering Capabilities

Some network-based IDPSs offer limited information gathering capabilities, which means that they can collect information on hosts and the network activity involving those hosts. Examples of information gathering capabilities are as follows:

- **Identifying Hosts.** An IDPS sensor might be able to create a list of hosts on the organization's network arranged by IP address or MAC address. The list can be used as a profile to identify new hosts on the network.
- **Identifying Operating Systems.** An IDPS sensor might be able to identify the OSs and OS versions used by the organization's hosts through various techniques. For example, the sensor could track which ports are used on each host, which could indicate a particular OS or OS family (e.g., Windows, Unix). Another technique is to analyze packet headers for certain unusual characteristics or combinations of characteristics that are exhibited by particular OSs; this is known as *passive fingerprinting*. Some sensors can also identify application versions (as described below), which in some cases implies which OS is in use. Knowing which OS versions are in use can be helpful in identifying potentially vulnerable hosts.
- **Identifying Applications.** For some applications, an IDPS sensor can identify the application versions in use by keeping track of which ports are used and monitoring certain characteristics of application communications. For example, when a client establishes a connection with a server, the server might tell the client what application server software version it is running, and vice versa. Information on application versions can be used to identify potentially vulnerable applications, as well as unauthorized use of some applications.
- **Identifying Network Characteristics.** Some IDPS sensors collect general information about network traffic related to the configuration of network devices and hosts, such as the number of hops between two devices. This information can be used to detect changes to the network configuration.

4.3.2 Logging Capabilities

Network-based IDPSs typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, to investigate incidents, and to correlate events between the IDPS and other logging sources. Data fields commonly logged by network-based IDPSs include the following:

- Timestamp (usually date and time)
- Connection or session ID (typically a consecutive or unique number assigned to each TCP connection or to like groups of packets for connectionless protocols)
- Event or alert type[21]
- Rating (e.g., priority, severity, impact, confidence)
- Network, transport, and application layer protocols
- Source and destination IP addresses
- Source and destination TCP or UDP ports, or ICMP types and codes
- Number of bytes transmitted over the connection
- Decoded payload data, such as application requests and responses
- State-related information (e.g., authenticated username)

[21] In the console, the event or alert type often links to supporting information for the specific vulnerability or exploit, such as references for additional information and associated Common Vulnerabilities and Exposures (CVE) numbers.

- Prevention action performed (if any).

Most network-based IDPSs can also perform packet captures. Typically this is done once an alert has occurred, either to record subsequent activity in the connection or to record the entire connection if the IDPS has been temporarily storing the previous packets.

4.3.3 Detection Capabilities

Network-based IDPSs typically offer extensive and broad detection capabilities. Most products use a combination of signature-based detection, anomaly-based detection, and stateful protocol analysis techniques to perform in-depth analysis of common protocols; organizations should use network-based IDPS products that use such a combination of techniques. The detection methods 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. Some products also use the same techniques and provide the same functionality as network behavior analysis (NBA) software; see Section 6 for additional information.

This section discusses the following aspects of detection capabilities:

- Types of events detected
- Detection accuracy
- Tuning and customization
- Technology limitations.

4.3.3.1 Types of Events Detected

The types of events most commonly detected by network-based IDPS sensors include the following:

- **Application layer reconnaissance and attacks** (e.g., banner grabbing, buffer overflows, format string attacks, password guessing, malware transmission). Most network-based IDPSs analyze several dozen application protocols. Commonly analyzed ones include Dynamic Host Configuration Protocol (DHCP), DNS, Finger, FTP, HTTP,[22] Internet Message Access Protocol (IMAP), Internet Relay Chat (IRC), Network File System (NFS), Post Office Protocol (POP), rlogin/rsh, Remote Procedure Call (RPC), Session Initiation Protocol (SIP), Server Message Block (SMB), SMTP, SNMP, Telnet, and Trivial File Transfer Protocol (TFTP), as well as database protocols, instant messaging applications, and peer-to-peer file sharing software.
- **Transport layer reconnaissance and attacks** (e.g., port scanning, unusual packet fragmentation, SYN floods). The most frequently analyzed transport layer protocols are TCP and UDP.
- **Network layer reconnaissance and attacks** (e.g., spoofed IP addresses, illegal IP header values). The most frequently analyzed network layer protocols are IPv4, ICMP, and IGMP. Many products are also adding support for IPv6 analysis. The level of IPv6 analysis that network-based IDPSs can

[22] Although network-based IDPSs can analyze HTTP protocol activity, they usually cannot perform analysis on the use of Web services, such as Simple Object Access Protocol (SOAP) and Extensible Markup Language (XML) messages carried over HTTP. Security technologies known as XML gateways or XML firewalls have been created specifically to analyze Web services activity. In addition to providing intrusion prevention functions, these technologies also perform firewalling, authentication and authorization services, access control, and audit logging. More information on XML gateways is available from NIST SP 800-95, *Guide to Web Services Security (DRAFT)*, which is available at http://csrc.nist.gov/publications/drafts.html.

perform varies considerably among products. Some products provide no IPv6 support or can simply alert administrators that IPv6 activity is present. Other products can do basic processing of IPv6 and tunneled IPv6 traffic, such as recording source and destination IP addresses, and extracting payloads (e.g., HTTP, SMTP) for in-depth analysis. Some products can do full analysis of the IPv6 protocol, such as confirming the validity of IPv6 options, to identify anomalous use of the protocol. Organizations with a current or future need to monitor IPv6 activity should carefully evaluate the IPv6 analysis capabilities of network-based IDPS products.[23]

- **Unexpected application services** (e.g., tunneled protocols, backdoors, hosts running unauthorized application services). These are usually detected through stateful protocol analysis methods, which can determine if the activity in a connection is consistent with the expected application protocol, or through anomaly detection methods, which can identify changes in network flows and open ports on hosts.
- **Policy violations** (e.g., use of inappropriate Web sites, use of forbidden application protocols). Some types of security policy violations can be detected by IDPSs that allow administrators to specify the characteristics of activity that should not be permitted, such as TCP or UDP port numbers, IP addresses, Web site names, and other pieces of data that can be identified by examining network traffic.

Some IDPSs can also monitor the initial negotiation conducted when establishing encrypted communications to identify client or server software that has known vulnerabilities or is misconfigured. This can include application layer protocols such as secure shell (SSH) and Secure Sockets Layer (SSL), and network layer virtual private networking protocols such as IP Security (IPsec).

Network-based IDPS sensors can often determine if an attack is likely to succeed. For example, as described in Section 4.3.3.3, sensors might know which Web server software versions are running on each of the organization's Web servers. If an attacker launches an attack against a Web server that is not vulnerable to the attack, then the sensor might produce a low-priority alert; if the server is thought to be vulnerable, then the sensor might produce a high-priority alert. IDPS sensors are typically configured to stop attacks whether or not they are likely to succeed, but an IDPS might still log the activity with different priority levels depending on what its outcome probably would have been if not blocked.

4.3.3.2 Detection Accuracy

Historically, network-based IDPSs have been associated with high rates of false positives and false negatives. Most of the early technologies relied primarily on signature-based detection, which by itself is accurate only for detecting relatively simple well-known threats. Newer technologies use a combination of detection methods to increase accuracy and the breadth of detection, and generally the rates of false positives and false negatives have declined. Another common problem with network-based IDPSs' accuracy is that they typically require considerable tuning and customization to take into account the characteristics of the monitored environment.

False positives and false negatives for network-based IDPS sensors can only be reduced somewhat because of the complexity of the activities being monitored. A single sensor is often monitoring traffic involving hundreds or thousands of internal and external hosts. The number and variety of OSs and applications in use over the monitored network can be immense; also, OSs and applications are constantly being changed. This makes it impossible for a sensor to understand everything it sees.

[23] NIST SP 500-267, *A Profile for IPv6 in the U.S. Government, Version 1.0 (DRAFT)*, provides recommendations for the IPv6-handling capabilities of network-based IDPS products. The draft is available at http://www.antd.nist.gov/.

Even worse, sensors have to monitor activity for many different combinations of servers and clients. For example, an organization could use 10 different types and versions of Web servers, which users could access with 50 different types and versions of Web browsers. Each combination of browser and server could have unique communication characteristics (e.g., sequence of commands, response codes) that could impact the accuracy of analysis. Also, different configurations and customizations could be applied to the browsers and servers. Security controls between the servers and clients that alter network activity, such as firewalls and proxy servers, could cause additional difficulties for sensors.

Ideally, network-based IDPSs would be able to interpret all network activity just as the endpoints do. For example, different types of Web servers can interpret the same Web request in different ways. Stateful protocol analysis techniques often attempt to do this by replicating the processing performed by common types of clients and servers. This allows the sensors to improve their detection accuracy slightly. Many attackers employ client and server-specific processing characteristics, such as handling character encodings, in their attacks as evasion techniques. Organizations should use network-based IDPSs that can compensate for the use of common evasion techniques.

4.3.3.3 Tuning and Customization

As mentioned in Section 4.3.3.2, network-based IDPSs usually require extensive tuning and customization to improve their detection accuracy. Examples of tuning and customization capabilities are thresholds for port scans and application authentication attempts, blacklists and whitelists for host IP addresses and usernames, and alert settings. Some products also provide code editing features, which is usually limited to signatures but in some cases may allow access to additional code, such as programs used to perform stateful protocol analysis.

Some network-based IDPSs can use information regarding the organization's hosts 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.

4.3.3.4 Technology Limitations

Although network-based IDPSs offer extensive detection capabilities, they do have some significant limitations. Three of the most important are analyzing encrypted network traffic, handling high traffic loads, and withstanding attacks against the IDPSs themselves. These limitations are discussed below.

Network-based IDPSs cannot detect attacks within encrypted network traffic, including virtual private network (VPN) connections, HTTP over SSL (HTTPS), and SSH sessions. As previously mentioned, some network-based IDPSs can do some analysis of the setup of encrypted connections, which can identify that the client or server software has known vulnerabilities or is misconfigured. To ensure that sufficient analysis is performed on payloads within encrypted network traffic, organizations should use IDPSs that can analyze the payloads before they are encrypted or after they are decrypted. Examples include placing network-based IDPS sensors to monitor unencrypted traffic (e.g., traffic that entered an organization through a VPN gateway and has since been decrypted) 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. Passive IDPS sensors might drop some packets, which could cause some incidents to go undetected, especially if stateful protocol analysis methods are in use. For inline IDPS sensors, dropping packets under high loads causes disruptions in network availability; also, delays in processing packets could cause unacceptably high latency. To avoid this, organizations using inline IDPS sensors should select ones that can recognize high load conditions and either pass certain types of network traffic through the sensor without performing full analysis (i.e., partial or no analysis) or drop low-priority traffic to reduce load. Many vendors attempt to optimize their sensors to provide better performance under high loads by taking measures such as using specialized hardware (e.g., high-bandwidth network cards) and recompiling components of their software to incorporate settings and other customizations made by administrators. Although vendors typically rate their sensors by maximum bandwidth capability, the actual capacity of any product depends on several factors, including the following:

- The network, transport, and application layer protocols in use, and the depth of analysis performed for each protocol. Vendors often rate their products based on their ability to perform reasonable analysis of a "typical" mix of protocols. The level of analysis that an individual organization wants to perform and the organization's mix of protocols may vary significantly from the tested conditions.
- The longevity of connections. For example, a sensor might have less overhead for one long-term connection than several consecutive short-term connections.
- The number of simultaneous connections. Sensors usually are limited as to how many connections for which they can track state.

IDPS sensors are susceptible to various types of attacks. Attackers can generate unusually large volumes of traffic, such as distributed denial of service (DDoS) attacks, and anomalous activity (e.g., unusually fragmented packets) to attempt to exhaust a sensor's resources or cause it to crash. Another attack technique, known as *blinding*, generates network traffic that is likely to trigger many IDPS alerts in a short period of time; typically, the network traffic is specially crafted to take advantage of typical configurations of IDPS sensors. 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. The attacker's goal is that the blinding traffic will either cause the IDPS to fail in some way or generate so many alerts that the alerts for the real attack will go unnoticed. Many IDPS sensors can recognize the use of common DDoS and blinding tools and techniques; the sensors can alert administrators to the attack and then ignore the rest of the activity, reducing the load on the sensors. Organizations should select products that offer features that make them resistant to failure due to attack.

4.3.4 Prevention Capabilities

Network-based IDPS sensors offer various prevention capabilities, including the following (grouped by sensor type):

- **Passive Only**
  - **Ending the Current TCP Session.** A passive sensor can attempt to end an existing TCP session by sending TCP reset packets to both endpoints; this is sometimes called *session sniping*.[24] The sensor does this to make it appear to each endpoint that the other endpoint is trying to end the connection. The goal is for one of the endpoints to terminate the connection before an attack can succeed. Unfortunately, in many cases the reset packets are not received in time because the

[24] An inline sensor could potentially use this technique, but it is much weaker than other methods that inline sensors can perform, so in practice session sniping is rarely used on inline sensors.

attack traffic has to be monitored and analyzed, the attack detected, and the packets sent across networks to the endpoints. Also, since this technique is only applicable to TCP, it cannot be used for attacks carried in other types of packets, including UDP and ICMP. Session sniping is not widely used any more because other, newer prevention capabilities are more effective.

- **Inline Only**
  - **Performing Inline Firewalling.** Most inline IDPS sensors offer firewall capabilities that can be used to drop or reject suspicious network activity.
  - **Throttling Bandwidth Usage.** If a particular protocol is being used inappropriately, such as for a DoS attack, malware distribution, or peer-to-peer file sharing, some inline IDPS sensors can limit the percentage of network bandwidth that the protocol can use. This prevents the activity from negatively impacting bandwidth usage for other resources.
  - **Altering Malicious Content.** As described in Section 2.2, some inline IDPS sensors can sanitize part of a packet, which means that malicious content is replaced with benign content and the sanitized packet sent to its destination. A sensor that acts as a proxy might perform automatic normalization of all traffic, such as repackaging application payloads in new packets. This has the effect of sanitizing some attacks involving packet headers and some application headers, whether or not the IDPS has detected an attack. Some sensors can also strip infected attachments from e-mails and remove other discrete pieces of malicious content from network traffic.
- **Both Passive and Inline**
  - **Reconfiguring Other Network Security Devices.** Many IDPS sensors can instruct network security devices such as firewalls, routers, and switches to reconfigure themselves to block certain types of activity or route it elsewhere. This can be helpful in several situations, such as keeping an external attacker out of a network and quarantining an internal host that has been compromised (e.g., moving it to a quarantine VLAN). This prevention technique is useful only for network traffic that can be differentiated by packet header characteristics typically recognized by network security devices, such as IP addresses and port numbers.
  - **Running a Third-Party Program or Script.** Some IDPS sensors can run an administrator-specified script or program when certain malicious activity is detected. This could trigger any prevention action desired by the administrator, such as reconfiguring other security devices to block the malicious activity. Third-party programs or scripts are most commonly used when the IDPS does not support the prevention actions that administrators want to have performed.

Most IDPS sensors 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 prevention capability should be used. Some IDPS 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 prevention capabilities' configuration before enabling them, which reduces the risk of inadvertently blocking benign activity.

4.4 Management

Most network-based IDPS products offer similar management capabilities. This section discusses major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently.

4.4.1 Implementation

Once a network-based IDPS product has been selected, the administrators need to design an architecture, perform IDPS component testing, secure the IDPS components, and then deploy them. The following items list additions to the material presented in Section 3.3.1:

- **Architecture Design.** A consideration specific to network-based IDPSs is where the sensors should be placed on the network, which includes deciding how many sensors are needed, which sensors should be inline and which should be passive, and how passive sensors should be connected to the network (e.g., IDS load balancer, network tap, switch spanning port).
- **Component Testing and Deployment.** Implementing a network-based IDPS can necessitate brief network outages, particularly when deploying inline sensors. However, passive sensor deployment can also cause outages for several reasons, including installation of network taps and IDS load balancers, and reconfiguration of switches to activate spanning port functions.
- **Securing the IDPS Components.** Administrators should ensure that for both passive and inline sensors, IP addresses are not assigned to the network interfaces used to monitor network traffic, except for network interfaces also used for IDPS management. Operating a sensor without IP addresses assigned to its monitoring interfaces is known as operating in *stealth mode*. Stealth mode improves the security of the IDPS sensors because it prevents other hosts from initiating connections to them. This conceals the sensors from attackers and thus limits their exposure to attacks. However, attackers may be able to identify the existence of an IDPS 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.

4.4.2 Operation and Maintenance

The operation and maintenance of network-based IDPSs is performed in the same manner as that documented in the general information provided in Section 3.3.2.

4.5 Summary

A network-based IDPS monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity. Network-based IDPS components are similar to other types of IDPS technologies, except for the sensors. A network-based IDPS sensor monitors and analyzes network activity on one or more network segments. Sensors are available in two formats: appliance-based sensors, which are comprised of specialized hardware and software optimized for IDPS sensor use, and software-only sensors, which can be installed onto hosts that meet certain specifications.

Organizations should consider using management networks for their network-based IDPS deployments whenever feasible. If an IDPS is deployed without a separate management network, organizations should consider whether or not a VLAN is needed to protect the IDPS communications. In addition to choosing the appropriate network for the components, administrators also need to decide where the IDPS sensors should be located. Sensors can be deployed in one of two modes: inline sensors are deployed so that the network traffic they monitor must pass through them, while passive sensors are deployed so that they monitor copies of the actual network traffic. Generally, organizations should deploy inline sensors if prevention methods will be used and passive sensors if they will not.

Network-based IDPSs provide a wide variety of security capabilities. Some products can collect information on hosts such as which OSs they use and which application versions they use that communicate over networks. Network-based IDPSs can also perform extensive logging of data related to detected events; most can also perform packet captures. Network-based IDPSs usually offer extensive and broad detection capabilities. Most products use a combination of signature-based detection, anomaly-based detection, and stateful protocol analysis to perform in-depth analysis of common protocols; organizations should use network-based IDPS products that provide such a combination of detection features, because the combination increases detection accuracy. Organizations should also use network-based IDPSs that can compensate for the use of common evasion techniques, which further improves detection accuracy.

Network-based IDPSs have some significant limitations. They cannot detect attacks within encrypted network traffic; accordingly, either they should be deployed where they can monitor traffic before encryption or after decryption, or host-based IDPSs should be used on endpoints to monitor unencrypted activity. Network-based IDPSs are often unable to perform full analysis under high loads; organizations using inline sensors should select those that can recognize high load conditions and either pass certain types of traffic without performing full analysis or drop low-priority traffic to reduce load. Another limitation of network-based IDPSs is that they are susceptible to various types of attacks, most involving large volumes of traffic. Organizations should select products that offer features designed to make them resistant to failure due to attack. Organizations should also ensure that IP addresses are not assigned to the network interfaces of passive or inline sensors used to monitor network traffic, except for network interfaces used for both traffic monitoring and IDPS management.

Network-based IDPS sensors offer various prevention capabilities. Many passive sensors can attempt to end TCP sessions by resetting them, but this technique often does not work in time, and it is not applicable to non-TCP sessions, such as UDP and ICMP. Inline sensor-specific techniques include performing inline firewalling, throttling bandwidth usage, and altering malicious content, all of which are helpful for certain circumstances. Both passive and inline sensors can reconfigure other network security devices; they can also run third-party programs or scripts to initiate additional prevention actions.

This page has been left blank intentionally.

5. Wireless IDPS

A *wireless IDPS* monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity involving the protocols themselves. This section provides a detailed discussion of wireless IDPS technologies. First, it contains a brief overview of wireless networking, which is background material for understanding the rest of the section. Next, it covers the major components of wireless IDPSs and explains the architectures typically used for deploying the components. It also examines the security capabilities of the technologies in depth, including the methodologies they use to identify and stop suspicious activity. The rest of the section discusses the management capabilities of the technologies, including recommendations for implementation and operation.

5.1 Wireless Networking Overview

Wireless networking enables devices with wireless capabilities to use computing resources without being physically connected to a network. The devices simply need to be within a certain distance (known as the *range*) of the wireless network infrastructure. A *wireless local area network (WLAN)* is a group of wireless networking nodes within a limited geographic area that is capable of exchanging data through radio communications. 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.

This section provides a brief introduction to wireless networking. Section 5.1.1 provides an overview of the most commonly used WLAN standards. Section 5.1.2 discusses the fundamental components of WLANs. Finally, Section 5.1.3 briefly examines the major threats against WLANs. This material is intended only to provide a high-level overview of wireless networking as background information for the wireless IDPS material in the rest of the section.[25]

5.1.1 WLAN Standards

Most WLANs use the Institute of Electrical and Electronics Engineers (IEEE) 802.11 family of WLAN standards.[26] The most commonly used WLAN radio transmission standards are IEEE 802.11b and IEEE 802.11g, which use the 2.4 gigahertz (GHz) band, and IEEE 802.11a, which uses the 5 GHz band. IEEE 802.11a, b, and g include security features known collectively as Wired Equivalent Privacy (WEP). Unfortunately, WEP has several well-documented security problems. To overcome these, IEEE 802.11i was created; it specifies security components that work in conjunction with IEEE 802.11a, b, and g.

Another set of WLAN standards has been created by a non-profit industry consortium of WLAN equipment and software vendors called the Wi-Fi Alliance.[27] While IEEE was working on finalizing the 802.11i standard, the Alliance created an interim solution called Wi-Fi Protected Access (WPA). Published in October 2002, WPA is essentially a subset of the draft IEEE 802.11i requirements available at that time. WPA provides stronger security for WLAN communications than WEP. In conjunction

---

[25] This publication does not address IDPS technologies for other forms of wireless networking, such as Bluetooth. Bluetooth IDPS products have just started to become available, and as of late 2006 they offer few capabilities (device detection, service enumeration, limited vulnerability scanning). It also does not address technologies based on the IEEE 802.11n WLAN standard, which as of late 2006 has not been finalized. It is expected that the recommendations in this section should generally be applicable to wireless IDPS technologies for IEEE 802.11n-based WLANs.

[26] For more information on the IEEE 802.11 standards and other aspects of wireless network security, see NIST SP 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i,* and NIST SP 800-48, *Wireless Network Security: 802.11, Bluetooth and Handheld Devices* (http://csrc.nist.gov/publications/nistpubs/index.html).

[27] For more information on the Wi-Fi Alliance, visit their Web site at http://www.wi-fi.org/.

with the ratification of the IEEE 802.11i amendment, the Wi-Fi Alliance introduced WPA2, its term for interoperable equipment that is capable of supporting IEEE 802.11i requirements. WPA2 offers stronger security controls than either WPA or WEP.

5.1.2 WLAN Components

IEEE 802.11 WLANs have two fundamental architectural components:

- **Station (STA).** A *STA* is a wireless endpoint device. Typical examples of STAs are laptop computers, personal digital assistants (PDA), mobile phones, and other consumer electronic devices with IEEE 802.11 capabilities.
- **Access Point (AP).**[28] An *AP* logically connects STAs with a *distribution system* (DS), which is typically an organization's wired infrastructure. The DS is the means by which STAs can communicate with the organization's wired LANs and external networks such as the Internet. Figure 5-1 shows an example of how APs, STAs, and DSs are related.

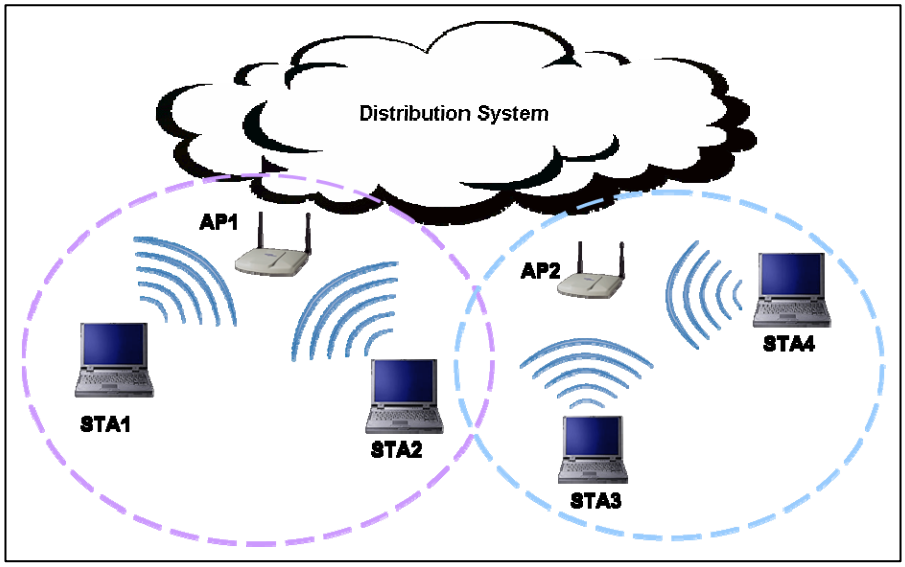

Figure 5-1. Wireless LAN Architecture Example

Some WLANs also use wireless switches. A *wireless switch* is a device that acts as an intermediary between APs and the DS. The purpose of the switch is to assist administrators in managing the WLAN infrastructure. In WLANs without wireless switches, the APs connect directly to the DS.

The IEEE 802.11 standard also defines the following two WLAN architectures:

- **Ad Hoc Mode.** The *ad hoc mode* does not use APs. Ad hoc mode, also known as peer-to-peer mode, involves two or more STAs communicating directly with one another.
- **Infrastructure Mode.** In *infrastructure mode*, an AP connects wireless STAs to a DS, typically a wired network.

[28] Technically, APs are also STAs. Some literature distinguishes between AP STAs and non-AP STAs. In this document, the term STA refers to non-AP STAs only.

Each AP and STA on a WLAN can be identified by its *media access control (MAC) address*, which is a unique 48-bit value that is assigned to a wireless network interface card. Part of the MAC address can be used to identify the card's vendor; the rest of the address acts as a serial number from the vendor. Ideally, MAC addresses could be used to uniquely identify every wireless device; however, it is relatively trivial to spoof a MAC address.

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. SSIDs are broadcast in plaintext by APs, so any listening wireless device can easily learn the SSID for each WLAN in its range.[29] Organizations may have no WLANs, one WLAN, or multiple WLANs. Also, many organizations have facilities that are within range of other organizations' WLANs.

5.1.3 Threats against WLANs

Although wireless and wired networks face the same general types of threats, the relative risk of some threats varies significantly. For example, wireless attacks typically require the attacker or a device placed by the attacker to be within close physical proximity to the wireless network; many attacks on wired networks can be performed remotely from any location. However, many WLANs are configured so that they do not require any authentication or require only weak forms of authentication; this makes it much easier for local attackers to perform several types of attacks, such as a man-in-the-middle attack.

Most WLAN threats involve an attacker with access to the radio link between a STA and an AP (or between two STAs, in ad hoc mode). Many attacks rely on an attacker's ability to intercept network communications or inject additional messages into them. This highlights the most significant difference between protecting wireless and wired LANs: the relative ease of accessing and altering network communications. In a wired LAN, an attacker would have to gain physical access to the LAN or remotely compromise systems on the LAN; in a wireless LAN, an attacker simply needs to be within range of the WLAN infrastructure.[30]

5.2 Components and Architecture

This section describes the major components of typical wireless IDPS solutions and illustrates the most common network architectures for these components. It also provides recommendations for the placement of certain components.

5.2.1 Typical Components

The typical components in a wireless IDPS are the same as a network-based IDPS: consoles, database servers (optional), management servers, and sensors. All of the components except sensors have essentially the same functionality for both types of IDPSs. Wireless sensors perform the same basic role as network-based IDPS sensors, but they 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 GHz and 5 GHz), and each

[29] Two WLANs within range of each other could have the same SSID. In such a case, the WLANs could be distinguished through the MAC addresses of their APs.

[30] More details on threats against WLANs are available from NIST SP 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i* and NIST SP 800-48, *Wireless Network Security: 802.11, Bluetooth and Handheld Devices* (http://csrc.nist.gov/publications/nistpubs/index.html).

band is separated into channels.[31] It is not currently possible for a sensor to monitor all traffic on a band simultaneously; a sensor has to monitor a single channel at a time. When the sensor is ready to monitor a different channel, the sensor must shut its radio off, change the channel, then turn its radio on. 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*, so that they can monitor each channel a few times per second. To reduce or eliminate channel scanning, specialized sensors are available that use several radios and high-power antennas, with each radio/antenna pair monitoring a different channel. 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.[32]

Wireless sensors are available in multiple forms:

- **Dedicated.** A dedicated sensor is a device that performs wireless IDPS functions but does not pass network traffic from source to destination. Dedicated sensors are often completely passive, functioning in a radio frequency (RF) monitoring mode to sniff wireless network traffic. Some dedicated sensors perform analysis of the traffic they monitor, while other sensors forward the network traffic to a management server for analysis. The sensor is typically connected to the wired network (e.g., Ethernet cable between the sensor and a switch). Dedicated sensors are usually designed for one of two deployment types:
  - **Fixed**—the sensor is deployed to a particular location. Such sensors are typically dependent on the organization's infrastructure (e.g., power, wired network).[33] Fixed sensors are usually appliance-based.
  - **Mobile**—the sensor is designed to be used while in motion. For example, a security administrator could use a mobile sensor while walking through an organization's buildings and campus to find rogue APs. Mobile sensors are either appliance-based or software-based (e.g., software installed onto a laptop with a wireless NIC capable of doing RF monitoring).[34]
- **Bundled with an AP.** Several vendors have added IDPS capabilities to APs. A bundled AP typically provides a less rigorous detection capability than a dedicated sensor because the AP needs to divide its time between providing network access and monitoring multiple channels or bands for

---

[31] IEEE 802.11b and g support 14 channels: 11 authorized for U.S. use and 3 authorized for international use. IEEE 802.11a supports 12 channels authorized for U.S. use and 4 channels authorized for international use. Some attackers use unusual channels or non-IEEE 802.11 frequency bands, such as 900 MHz or 4.9 GHz, because their activity is less likely to be detected than the use of the typical WLAN frequencies and channels. For example, an attacker who can gain unauthorized physical access to a wired network could install a wireless device that can subsequently transmit information from the organization to the attacker over an atypical frequency. Spectrum analyzer products can monitor activity on different frequency bands to identify attacks and to find benign sources of interference, such as cordless phones and microwaves. As of mid-2006, few IDPS products offer any spectrum analysis capabilities. However, several companies offer mobile handheld spectrum analyzers that can monitor common bands. A detailed discussion of them is outside the scope of this document.

[32] Organizations need to determine which channels should be monitored. As previously explained, attackers often use unusual channels, such as IEEE 802.11a, b, or g channels that are not authorized for U.S. use. Although monitoring these channels can detect such activity, it could reduce the percentage of time monitoring the channels that are used both by the organization's WLANs and also typical rogue WLANs, such as unauthorized access points. Organizations should consider the likelihood of the possible threats and choose a channel scanning plan that best addresses the threats.

[33] Some sensors can use the IEEE 802.3af protocol, also known as Power over Ethernet (PoE). This allows a sensor to receive its electrical power through the same Ethernet cable that connects it to the wired network. PoE is implemented in some dedicated sensors and access points. More information on PoE is available at http://www.ieee802.org/3/af/index.html

[34] Mobile sensors may be part of an enterprise wireless IDPS solution or may be standalone devices, managed and monitored directly by an administrator.

malicious activity. If the IDPS only needs to monitor a single band and channel, a bundled solution might provide reasonable security and network availability. If the IDPS has to monitor multiple bands or channels, then the sensor needs to perform channel scanning, which will disrupt the AP functions of the sensor by making it temporarily unavailable on its primary band and channel.

- **Bundled with a Wireless Switch.** Wireless switches are intended to assist administrators with managing and monitoring wireless devices; some of these switches also offer some wireless IDPS capabilities as a secondary function. Wireless switches typically do not offer detection capabilities as strong as bundled APs or dedicated sensors.

Because dedicated sensors can focus on detection and do not need to carry wireless traffic, they typically offer stronger detection capabilities than wireless sensors bundled with APs or wireless switches. However, dedicated sensors are often more expensive to acquire, install, and maintain than bundled sensors because bundled sensors can be installed on existing hardware, whereas dedicated sensors involve additional hardware and software. Organizations should consider both security and cost when selecting wireless IDPS sensors.

Some vendors also have host-based wireless IDPS sensor software that can be installed on STAs, such as laptops. The sensor software detects attacks within range of the STAs, as well as misconfigurations of the STAs, and reports this information to management servers. The sensor software may also be able to enforce security policies on the STAs, such as limiting access to wireless interfaces. More information on host-based IDPS products is presented in Section 7.

5.2.2 Network Architectures

Wireless IDPS components are typically connected to each other through a wired network, as shown in Figure 5-2. As with a network-based IDPS, a separate management network or the organization's standard networks can be used for wireless IDPS component communications. Because there should already be a strictly controlled separation between the wireless and wired networks, using either a management network or a standard network should be acceptable for wireless IDPS components. Also, some wireless IDPS sensors (particularly mobile ones) are used standalone and do not need wired network connectivity.

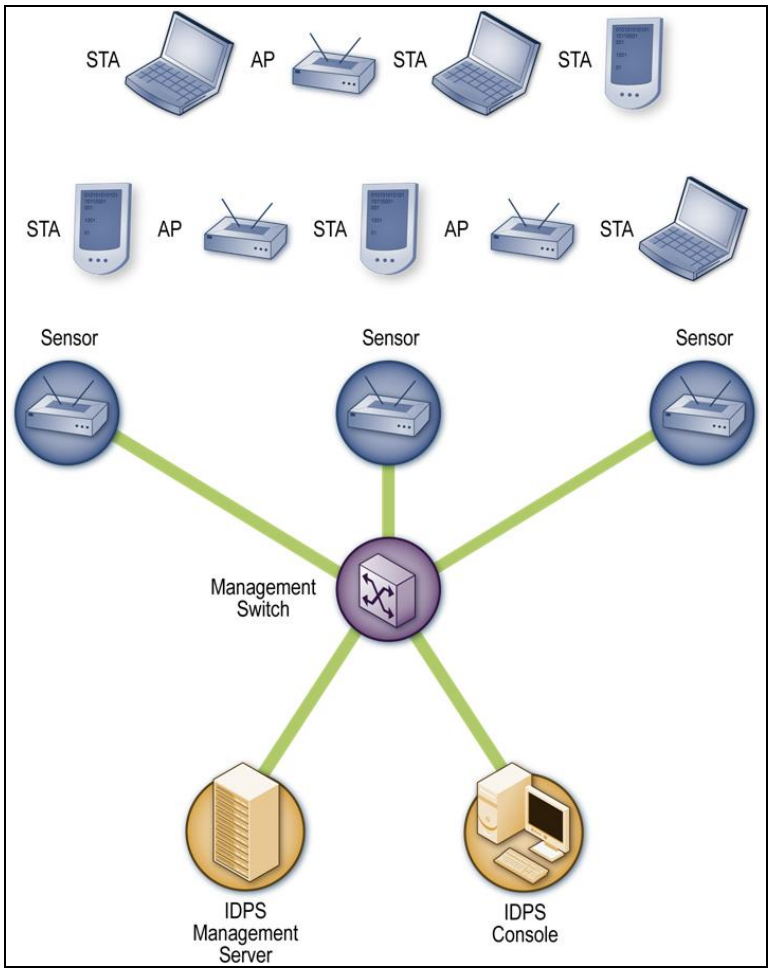

Figure 5-2. Wireless IDPS Architecture

5.2.3 Sensor Locations

Choosing sensor locations for a wireless IDPS deployment is a fundamentally different problem than choosing locations for any other type of IDPS sensor. If the organization uses WLANs, wireless sensors should be deployed so that they monitor the RF range of the organization's WLANs (both APs and STAs), which often includes mobile components such as laptops and PDAs. Many organizations also want to deploy sensors to monitor physical regions 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 APs and ad hoc WLANs. Other considerations for selecting wireless sensor locations include the following:

- **Physical Security.** Sensors are often deployed into open locations (e.g., hallway ceilings, conference rooms) because their range is much greater there than in closed locations (e.g., wiring closets). Sensors are sometimes deployed outdoors as well.[35] Generally, sensors in open interior locations and external locations are more susceptible to physical threats than other sensors. If the physical threats are significant, organizations might need to select sensors with anti-tamper features or deploy sensors where they are less likely to be physically accessed (e.g., within view of a security camera).

[35] Special sensors are available for outdoor use that offer better resistance to environmental threats than regular sensors.

- **Sensor Range.** The actual range of a sensor varies based on the surrounding facilities (e.g., walls, doors). Some wireless IDPS vendors offer modeling software that can analyze building floor plans and the attenuation characteristics of walls, doors, and other facility components to determine effective locations for sensors. Sensor range can also vary based on the location of people within the facility and other changing characteristics, so sensors should be deployed so that their ranges have some overlap (e.g., at least 20%).
- **Wired Network Connections.** The sensors typically need to be connected to the wired network. If there is a need to deploy sensors in an area where there is no wired network, then it might be necessary to extend the wired network into that area. This is generally a concern only if the organization wants to monitor portions of their facilities that are outside the range of the WLAN.
- **Cost.** Ideally, an organization could deploy sensors throughout its facilities to perform full wireless monitoring. However, the number of sensors needed to do so can be quite large, especially in wide-open campus environments. Organizations should compare WLAN threats to the cost of sensor purchases, deployment, and maintenance, and develop a solution that creates an acceptable level of risk. For example, an organization might decide to deploy fixed sensors throughout the range of the organization's WLANs, and to do periodic checks of other areas using mobile sensors.
- **AP and Wireless Switch Locations.** If a bundled solution (e.g., wireless IDPS software on an AP) would meet the organization's other requirements, then the locations of APs and wireless switches are particularly important because the wireless IDPS software could potentially be deployed onto those devices.

5.3 Security Capabilities

Wireless IDPSs provide several types of security capabilities. Because wireless IDPS is a relatively new form of IDPS, capabilities currently vary widely among products; over time, product capabilities should become more consistent. Sections 5.3.1 through 5.3.4 describe common security capabilities, divided into four categories: information gathering, logging, detection, and prevention, respectively.

5.3.1 Information Gathering Capabilities

Most wireless IDPSs can collect information on wireless devices. Examples of these information gathering capabilities are as follows:

- **Identifying WLAN Devices.** Most IDPS sensors can create and maintain an inventory of observed WLAN devices, including APs, WLAN clients, and ad hoc (peer-to-peer) clients. The inventory is usually based on SSIDs and the MAC addresses of the devices' wireless network cards; the first portion of each MAC address identifies the vendor of the card.[36] Some sensors can also use fingerprinting techniques on observed traffic to verify the vendor, instead of relying on the MAC information (which could be spoofed). The inventory can be used as a profile to identify new WLAN devices and the removal of existing devices.
- **Identifying WLANs.** Most IDPS sensors keep track of observed WLANs, identifying them by their SSIDs. Administrators can then tag each entry as being an authorized WLAN, a benign neighboring WLAN (e.g., another organization in the same building), or a rogue WLAN. This information can be used to identify new WLANs, as well as to prioritize responses to identified events.

[36] Some STAs transmit multiple SSIDs while trying to identify previously accessed WLANs.

5.3.2 Logging Capabilities

Wireless IDPSs typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, to investigate incidents, and to correlate events between the IDPS and other logging sources. Data fields commonly logged by wireless IDPSs include the following:

- Timestamp (usually date and time)
- Event or alert type[37]
- Priority or severity rating
- Source MAC address (the vendor is often identified from the address)
- Channel number
- ID of the sensor that observed the event
- Prevention action performed (if any).

5.3.3 Detection Capabilities

Wireless IDPSs can detect attacks, misconfigurations, and policy violations at the WLAN protocol level, primarily examining IEEE 802.11a, b, g, and i 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 detection, anomaly-based detection, and stateful protocol analysis techniques; organizations should use wireless IDPS products that use such a combination of techniques, to achieve broader and more accurate detection. This section discusses the following aspects of detection capabilities:

- Types of events detected
- Detection accuracy
- Tuning and customization
- Technology limitations.

5.3.3.1 Types of Events Detected

The types of events most commonly detected by wireless IDPS sensors include the following:

- **Unauthorized WLANs and WLAN devices**. Through their information gathering capabilities, most wireless IDPS sensors can detect rogue APs, unauthorized STAs, and unauthorized WLANs (both infrastructure mode and ad hoc mode).
- **Poorly secured WLAN devices**. Most wireless IDPS sensors can identify APs and STAs that are not using the proper security controls. This includes detecting misconfigurations and the use of weak WLAN protocols and protocol implementations. This is accomplished by identifying deviations from organization-specific policies for settings such as encryption, authentication, data rates, SSID names, and channels. For example, a sensor could detect that a STA is using WEP instead of WPA2 or IEEE

[37] Some products may use identifiers from the Wireless Vulnerabilities & Exploits (WVE) database, which contains information on wireless protocol and product vulnerabilities and known exploits for these vulnerabilities. The WVE database is located at http://www.wve.org/.

802.11i. The majority of types of events that can be detected by wireless IDPSs fall into this detection category.

- **Unusual usage patterns**. Some sensors can use anomaly-based detection methods to detect unusual WLAN usage patterns. For example, if many more STAs than usual are using a particular AP, or there is a much higher than usual amount of network traffic between a STA and AP, one of the devices might have been compromised, or unauthorized parties might be using the WLAN. Many sensors can identify failed attempts to join the WLAN, such as alerting on several failed attempts in a short period of time, which could indicate an attempt to gain unauthorized access to the WLAN. Some sensors can also alert if any WLAN activity is detected during off-hours periods.

- **The use of wireless network scanners** (e.g., war driving tools). Such scanners are used to identify unsecured or weakly secured WLANs. Wireless IDPS sensors can detect only the use of active scanners—scanners that generate wireless network traffic. They cannot detect the use of passive sensors that simply monitor and analyze observed traffic.[38]

- **Denial of service (DoS) attacks and conditions** (e.g., network interference). DoS attacks include logical attacks such as *flooding*, which involves sending large numbers of messages to an AP at a high rate, and physical attacks such as *jamming*, which involves emitting electromagnetic energy on the WLAN's frequencies to make the frequencies unusable by the WLAN. DoS attacks can often be detected through stateful protocol analysis and anomaly detection methods, which can determine if the observed activity is consistent with the expected activity. Many denial of service attacks are detected by counting events during periods of time and alerting when threshold values are exceeded. For example, a large number of events involving the termination of wireless network sessions can indicate a DoS attack.

- **Impersonation and man-in-the-middle attacks**. Some wireless IDPS sensors can detect when a device is attempting to spoof the identity of another device. This is done by identifying differences in the characteristics of the activity, such as certain values in frames.

Most wireless IDPS sensors can identify the physical location of a detected threat by using *triangulation*—estimating the threat's approximate distance from multiple sensors by the strength of the threat's signal received by each sensor, then calculating the physical location at which the threat would be the estimated distance from each sensor. This allows an organization to send physical security staff to the location to address the threat. Wireless IDPS products that can use building floor plans can also determine if the threat is inside or outside a building, or if it is in a public area or secured area. This information is helpful not only in finding and stopping the threat, but also in prioritizing the response to the threat. Wireless IDPS sensors can set the priority of alerts based in part on the location of each threat. Handheld IDPS sensors can also be used to pinpoint a threat's location, particularly if fixed sensors do not offer triangulation capabilities or if the threat is moving.

5.3.3.2 Detection Accuracy

Compared to other forms of IDPS, wireless IDPS is generally more accurate; this is largely due to its limited scope (analyzing wireless networking protocols). 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 within the organization's facilities.

[38] In many cases, the most effective way to identify the use of passive scanners is through physical security controls, such as seeing individuals with computers and antennas in proximity to the organization's facilities.

5.3.3.3 Tuning and Customization

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., application), there are generally not a large number of alert types, and consequently not many customizations or tunings available. Some wireless IDPSs offer industry-specific templates that can be helpful in establishing base policies.

Wireless IDPSs offer some customization features. Most have thresholds that can be used for anomaly-based detection. Blacklists and whitelists are used to hold lists of known malicious and benign WLAN devices, respectively. The lists can also be used to record authorized or unauthorized WLAN NIC vendors; alerts can be generated when any NICs not on the authorized list are used for APs or STAs. Individual alerts can be customized, just as they can for network-based IDPSs. Code editing is not available for most products, although some vendors allow administrators to enter complex logical expressions to tune certain detection capabilities.

Besides reviewing tuning and customizations periodically to ensure that they are still accurate, administrators should also ensure that changes to building plans are incorporated occasionally. This is needed for accurate identification of the physical location of threats and accurate planning of sensor deployments.

5.3.3.4 Technology Limitations

Although wireless IDPSs offer robust detection capabilities, they do have some significant limitations. Three of the most important are being unable to detect certain wireless protocol attacks, being susceptible to evasion techniques, and being unable to withstand attacks against the IDPSs themselves. These limitations are discussed in detail below.

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., 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 that is collected from the same WLAN. Wireless IDPSs cannot fully compensate for the use of insecure wireless networking protocols.

Another problem with some wireless IDPS sensors is the use of evasion techniques. Attackers can identify the wireless IDPS product in use by various means, including a physical survey of the area in which the sensors are deployed, and the use of fingerprinting techniques that can identify the product in use by the characteristics of its prevention actions (see Section 5.3.4 for information on prevention). Once an attacker has identified the product, evasion techniques can be used that take advantage of the product's channel scanning scheme. 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 wireless IDPS sensor detects the first attack, it cannot detect the second attack unless it scans away from the channel of the first attack. Another drawback of channel scanning is the impact it could have on network forensics. Since each sensor sees only a fraction of the activity on each channel, the forensic data is quite incomplete, making it considerably more difficult to analyze.

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.

5.3.4 Prevention Capabilities

Wireless IDPS sensors offer two types of intrusion prevention capabilities:

- **Wireless.** Some sensors can terminate connections between a rogue or misconfigured STA and an authorized AP or between an authorized STA and a rogue or misconfigured AP through the air. This is typically done by sending messages to the endpoints, telling them to deassociate the current session. The sensor then refuses to permit a new connection to be established.
- **Wired.** Some sensors can instruct a switch on the wired network to block network activity involving a particular STA or AP based on the device's MAC address or switch port. For example, if a STA is sending attacks to a server on the wired network, a sensor could direct a wired switch to block all activity to and from the STA. This technique is only effective for blocking the malicious STA or AP's wired network communications. It will not stop a STA or AP from continuing to perform malicious actions through wireless protocols.

Most IDPS sensors 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 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 prevention capabilities' configuration before enabling prevention, which reduces the risk of performing prevention actions on benign activity.

An important consideration 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. When selecting sensors, organizations should consider what prevention actions may need to be performed and how the sensor's detection capabilities could be affected by performing prevention actions.

5.4 Management

Most wireless IDPS products offer similar management capabilities. This section discusses major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently.

5.4.1 Implementation

Once a wireless IDPS product has been selected, the administrators need to design an architecture, perform IDPS component testing, secure the IDPS components, and then deploy them. The only addition to the material presented in Section 3.3.1 involves component testing and deployment. Implementing a wireless IDPS can necessitate brief wireless network outages if existing APs or wireless switches need to be upgraded or have IDPS software installed. Generally, the deployment of dedicated sensors causes no network outages.

5.4.2 Operation and Maintenance

The operation and maintenance of a wireless IDPS solution is nearly identical to that of a network-based IDPS solution. Wireless IDPS consoles offer similar management, monitoring, analysis, and reporting capabilities. One significant difference is that wireless IDPS consoles can display the physical location of threats. A minor difference is that because wireless IDPS sensors detect a relatively small variety of events, compared to other types of IDPSs, they tend to have signature updates less frequently.

5.5 Summary

A wireless IDPS monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity. The typical components in a wireless IDPS are the same as a network-based IDPS: consoles, database servers (optional), management servers, and sensors. However, unlike a network-based IDPS sensor, which can see all packets on the networks it monitors, a wireless IDPS sensor works by sampling traffic because it can only 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, so that they can monitor each channel a few times per second.

Wireless sensors are available in multiple forms. A dedicated sensor is a fixed or mobile device that performs wireless IDPS functions but does not pass network traffic from source to destination. The other wireless sensor forms are bundled with access points (AP) or wireless switches. Because dedicated sensors can focus on detection and do not need to carry wireless traffic, they typically offer stronger detection capabilities than wireless sensors bundled with access points or wireless switches. However, dedicated sensors are often more expensive to acquire, install, and maintain than bundled sensors because bundled sensors can be installed on existing hardware, whereas dedicated sensors involve additional hardware and software. Organizations should consider both security and cost when selecting wireless IDPS sensors.

Wireless IDPS components are typically connected to each other through a wired network. Because there should already be a strictly controlled separation between the wireless and wired networks, using either a management network or a standard network should be acceptable for wireless IDPS components. Choosing sensor locations for a wireless IDPS deployment is a fundamentally different problem than choosing locations for any other type of IDPS sensor. If the organization uses wireless local area networks (WLAN), wireless sensors should be deployed so that they monitor the range of the WLANs. Many organizations also want to deploy sensors to monitor parts of their facilities where there should be no WLAN activity, as well as channels and bands that the organization's WLANs should not use. Other considerations for selecting sensor locations include physical security, sensor range, wired network connection availability, cost, and AP and wireless switch locations.

Wireless IDPSs provide several types of security capabilities. Most can collect information on observed wireless devices and WLANs and perform extensive logging of event data. Wireless IDPSs can detect attacks, misconfigurations, and policy violations at the WLAN protocol level. Organizations should use wireless IDPS products that use a combination of detection techniques to achieve broader and more accurate detection. Examples of events detected by wireless IDPSs are unauthorized WLANs and WLAN devices, poorly secured WLAN devices, unusual usage patterns, the use of active wireless network scanners, denial of service attacks, and impersonation and man-in-the-middle attacks. Most wireless IDPS sensors can also identify the physical location of a detected threat by using triangulation.

Compared to other forms of IDPS, wireless IDPS is generally more accurate; this is largely due to its limited scope (analyzing wireless networking protocols). Wireless IDPSs 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. Besides reviewing tuning and customizations periodically to ensure that they are still accurate, administrators should also ensure that changes to building plans are incorporated occasionally. This is needed for accurate identification of the physical location of threats and accurate planning of sensor deployments.

Although wireless IDPSs offer robust detection capabilities, they do have some significant limitations. Wireless IDPSs cannot detect certain types of attacks against wireless networks, such as attacks involving passive monitoring and offline processing of wireless traffic. Wireless IDPSs are also susceptible to evasion techniques, especially those involving knowledge of a product's channel scanning scheme. Channel scanning can also impact network forensics because each sensor sees only a fraction of the activity on each channel. Wireless IDPS sensors are also susceptible to denial of service attacks and physical attacks.

Wireless IDPS sensors can offer intrusion prevention capabilities. Some sensors can instruct endpoints to terminate a session and prevent a new session from being established. Some sensors can instruct a switch on the wired network to block network activity for a particular wireless endpoint; however, this method can only block wired network communications and will not stop an endpoint from continuing to perform malicious actions through wireless protocols. Most IDPS sensors allow administrators to specify the prevention capability configuration for each type of alert. Prevention actions can affect 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. When selecting sensors, organizations should consider what prevention actions may need to be performed and how the sensor's detection capabilities could be affected by performing prevention actions.

This page has been left blank intentionally.

6. Network Behavior Analysis (NBA) System

A network behavior analysis (NBA) system examines network traffic or statistics on network traffic to identify 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 network services to other systems).[39] This section provides a detailed discussion of NBA technologies. First, it covers the major components of the NBA technologies and explains the architectures typically used for deploying the components. It also examines the security capabilities of the technologies in depth, including the methodologies they use to identify suspicious activity. The rest of the section discusses the management capabilities of the technologies, including recommendations for implementation and operation.

6.1 Components and Architecture

This section describes the major components of typical NBA solutions and illustrates the most common network architectures for these components. It also provides recommendations for the placement of certain components.

6.1.1 Typical Components

NBA solutions usually have sensors and consoles, with some products also offering management servers (which are sometimes called *analyzers*). NBA sensors are usually available only as appliances. 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. 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. There are many standards for flow data formats, including NetFlow[40] and sFlow.[41] Typical flow data particularly relevant to intrusion detection and prevention includes the following:

- Source and destination IP addresses
- Source and destination TCP or UDP ports or ICMP types and codes
- Number of packets and number of bytes transmitted in the session
- Timestamps for the start and end of the session.

6.1.2 Network Architectures

As with a network-based IDPS, a separate management network or the organization's standard networks can be used for NBA component communications. If sensors that collect network flow data from other devices are used, the entire NBA solution can be logically separated from the standard networks. Figure 6-1 shows an example of an NBA network architecture.

---

[39] Some vendors refer to NBA technology as network behavior anomaly detection (NBAD) software, network behavior analysis and response software, or network anomaly detection software.

[40] More information on NetFlow is available from RFC 3954, *Cisco Systems NetFlow Services Export Version 9* (http://www.ietf.org/rfc/rfc3954.txt) and from the Cisco Web site at http://www.cisco.com/en/US/products/ps6645/products_ios_protocol_option_home.html.

[41] More information on sFlow is available at http://www.sflow.org/.

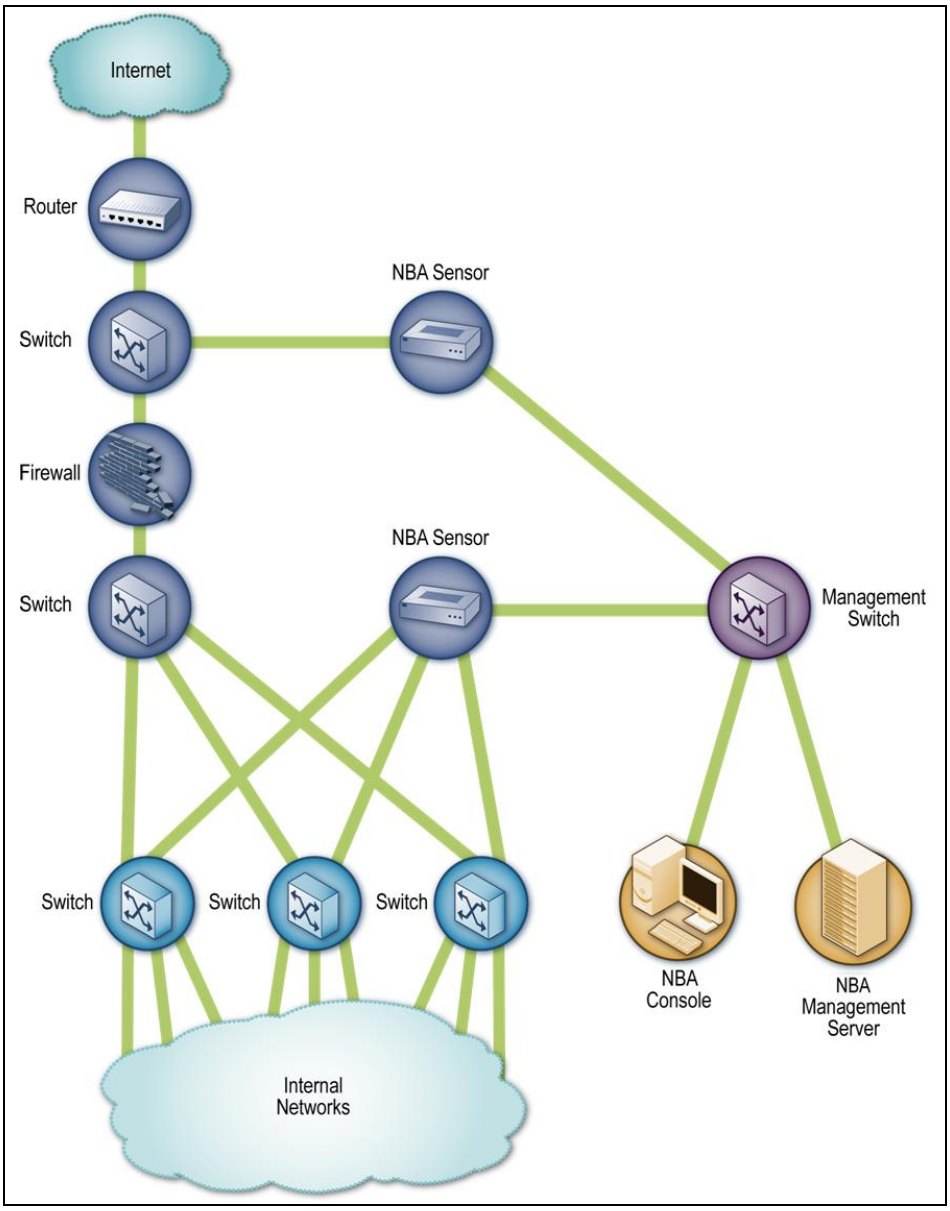

Figure 6-1. NBA Sensor Architecture Example

6.1.3 Sensor Locations

In addition to choosing the appropriate network for the components, administrators also need to decide where the sensors should be located. Most NBA sensors can be deployed in passive mode only, using the same connection methods (e.g., network tap, switch spanning port) as network-based IDPSs. Passive sensors that are performing direct network monitoring should be placed so that they can monitor key network locations, such as the divisions between networks, and key network segments, such as demilitarized zone (DMZ) subnets. Inline sensors are typically intended for network perimeter use, so they would be deployed in close proximity to the perimeter firewalls, often between the firewall and the Internet border router to limit incoming attacks that could overwhelm the firewall.

6.2 Security Capabilities

NBA products provide a variety of security capabilities. Sections 6.2.1 through 6.2.4 describe common security capabilities, divided into four categories: information gathering, logging, detection, and prevention, respectively. Some NBA products also provide security information and event management (SIEM) capabilities; see Section 8.2.2 for information on SIEM.

6.2.1 Information Gathering Capabilities

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. (As described in Section 6.2.3.1, most products also allow administrators to specify detailed firewall ruleset-like policies for host-to-host communications, including permitted or forbidden port numbers.) Information typically collected for each host includes the following:

- IP address
- Operating system
- What services it is providing, including the IP protocols and TCP and UDP ports it uses to do so
- Other hosts with which it communicates, and what services it uses and which IP protocols and TCP or UDP ports it contacts on each host.

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; this is discussed in Section 6.2.3.

6.2.2 Logging Capabilities

NBA technologies typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, to investigate incidents, and to correlate events between the NBA solution and other logging sources. Data fields commonly logged by NBA software include the following:

- Timestamp (usually date and time)
- Event or alert type
- Rating (e.g., priority, severity, impact, confidence)
- Network, transport, and application layer protocols
- Source and destination IP addresses
- Source and destination TCP or UDP ports, or ICMP types and codes
- Additional packet header fields (e.g., IP time-to-live [TTL])
- Number of bytes and packets sent by the source and destination hosts for the connection
- Prevention action performed (if any).

Some NBA sensors that directly monitor network traffic are able to log limited payload information from packets, such as authenticated user identifiers. This allows actions to be traced to specific user accounts.

6.2.3 Detection Capabilities

NBA technologies typically have the capability to detect several types of malicious activity. Most products use primarily anomaly-based detection, along with some stateful protocol analysis techniques, to analyze network flows. 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 threats. This section discusses the following aspects of NBA software detection capabilities:

- Types of events detected
- Detection accuracy
- Tuning and customization
- Technology limitations.

6.2.3.1 Types of Events Detected

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

- **Denial of service (DoS) attacks** (including distributed denial of service [DDoS] attacks). These attacks typically involve significantly increased bandwidth usage or a much larger number of packets or connections to or from a particular host than usual. By monitoring these characteristics, anomaly detection methods can determine if the observed activity is significantly different than the expected activity. Some NBA sensors are aware of the characteristics of common DoS tools and methods, which can help them to recognize the threats more quickly and prioritize them more accurately.
- **Scanning.** Scanning can be detected by atypical flow patterns at the application layer (e.g., banner grabbing), transport layer (e.g., TCP and UDP port scanning), and network layer (e.g., ICMP scanning).
- **Worms.** Worms spreading among hosts can be detected in more than one way. Some worms propagate quickly and use large amounts of bandwidth. Worms can also be detected because they can cause hosts to communicate with each other that typically do not, and they can also cause hosts to use ports that they normally do not use. Many worms also perform scanning; this can be detected as previously explained.
- **Unexpected application services** (e.g., tunneled protocols, backdoors, use of forbidden application protocols). These are usually detected through stateful protocol analysis methods, which can determine if the activity within a connection is consistent with the expected application protocol.
- **Policy violations.** Most NBA sensors allow administrators to specify detailed policies, such as which hosts or groups of hosts a particular system may or may not contact, and what types of activity are permissible only during certain hours or days of the week. Most sensors also detect many possible policy violations automatically, such as detecting new hosts or new services running on hosts, which could be unauthorized.

Most NBA sensors can reconstruct a series of observed events to determine the origin of a threat. 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 to other hosts.

6.2.3.2 Detection Accuracy

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 and if they do not violate the administrator-set policies (e.g., the attack uses common ports and protocols).

Detection accuracy also varies over time. 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. If a DoS attack starts slowly and increases in volume over time, it is likely to be detected by NBA sensors, but the point during the attack at which the NBA software detects it may vary considerably among NBA products. 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 a few hosts start using it, an NBA sensor is likely to detect this as anomalous. However, typically this would be a low-priority alert, and not reported as an attack, so it is debatable whether this can truly be considered a false positive. If a major service is moved from one host to another and a thousand hosts start using it one day, that might inadvertently trigger an alert.

6.2.3.3 Tuning and Customization

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, other than updating firewall ruleset-like policies that are offered by most products. Also, administrators might adjust thresholds periodically (e.g., how much additional bandwidth usage should trigger an alert) to take into account changes to the environment. Thresholds can often be set on a per-host basis or for administrator-defined groups of hosts. Most NBA products also offer whitelist and blacklist capabilities for hosts and services. Another common feature of NBA products is customization of each alert (e.g., specifying which prevention option it should trigger). Unlike network-based IDPSs, code editing features are generally not applicable to NBA products.

A few NBA products offer limited signature-based detection capabilities. The supported signatures tend to be very simple, and primarily look 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. For example, suppose that there is a DDoS attack that uses a flood of specially crafted HTTP traffic against a Web server. A firewall or router might not be able to block the attack without blocking all HTTP activity to the Web server, but an inline NBA sensor could be configured with a customized signature to block just the attack activity if it has a unique set of characteristics. On the other hand, an inline NBA sensor might be able to block the attack anyway because of its flow patterns.

Besides reviewing tuning and customizations periodically to ensure that they are still accurate, administrators should also ensure that significant changes to hosts, such as new hosts and new services, are reflected in NBA settings. Although it might not feasible to automatically link NBA systems with

change management systems, administrators could review change management records regularly and adjust host inventory information in the NBA to prevent false positives.

6.2.3.4 Technology Limitations

NBA technologies offer strong detection capabilities for certain types of threats, but they also have significant limitations. Some of these limitations are described in Section 6.2.3.2. 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; depending on the product's capabilities, network capacity, and administrator preferences, this could occur relatively frequently (e.g., every minute, every two minutes) or relatively infrequently (e.g., every 15 minutes, every 30 minutes). Because of this delay, attacks that occur quickly, such as malware infestations and 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. Therefore, to do direct monitoring instead of using flow data, organizations might have to purchase more powerful sensors and/or more sensors.

6.2.4 Prevention Capabilities

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

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

Most NBA sensors 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. Most NBA system implementations use prevention capabilities in a limited fashion or not at all because of false positives; blocking a single false positive could cause major

disruptions in network communications. Prevention capabilities are most often used for NBA sensors when blocking a specific known threat, such as a new worm.

6.3 Management

Most NBA products offer similar management capabilities. This section discusses major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently.

6.3.1 Implementation

Once an NBA product has been selected, the administrators need to design an architecture, perform NBA component testing, secure the NBA components, and then deploy them. The only addition to the material presented in Section 3.3.1 involves component testing and deployment. When NBA components are being deployed to production networks, organizations should typically install the sensors in a relatively short period of time, so that they can all build their inventories and generate their initial baselines at the same time. Detection accuracy is likely to be decreased during implementation and initial usage because the sensors will have substantially incomplete information about their environment until they have monitored it for days or weeks. Other than that, deployment of NBA sensors and consoles is essentially the same as it is for network-based IDPS sensors and consoles.

6.3.2 Operation and Maintenance

NBA products are designed to be operated and maintained through consoles, which typically have very similar capabilities to the consoles for network-based IDPSs. A key difference is that NBA consoles usually offer visualization tools that can display the flow of attacks through an organization's networks. These tools can show a user which hosts were affected by an attack, the sequence of hosts that an attack passed through, and the first host to be involved in the attack. Some NBA products also offer command-line interfaces.

Ongoing maintenance of NBA products is also very similar to that for network-based IDPSs. The primary exception is the application of updates. Because most NBA products do not use signatures, administrators only need to test and apply updates to the NBA software itself. Because NBA sensors are appliance-based, updating them usually involves replacing an existing CD and either rebooting the sensor or installing software from the CD. For NBA products that do have signature capabilities, administrators should also acquire, test, and apply signature updates in the same way that network-based IDPS signature updates are performed.

6.4 Summary

A network behavior analysis (NBA) system examines network traffic or statistics on network traffic to identify unusual traffic flows. NBA solutions usually have sensors and consoles, with some products also offering management servers. 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. Other NBA sensors do not monitor the networks directly, but instead rely on network flow information provided by routers and other networking devices.

Most NBA sensors can be deployed in passive mode only, using the same connection methods (e.g., network tap, switch spanning port) as network-based IDPSs. Passive sensors that are performing direct network monitoring should be placed so that they can monitor key network locations, such as the divisions between networks, and key network segments, such as DMZ subnets. Inline sensors are

typically intended for network perimeter use, so they would be deployed in close proximity to the perimeter firewalls, often in front to limit incoming attacks that could overwhelm the firewalls.

NBA products provide a variety of security capabilities. They offer extensive information gathering capabilities, collecting detailed information on each observed host and constantly monitoring network activity for changes to this information. NBA technologies typically perform extensive logging of data related to detected events. They also typically have the capability to detect several types of malicious activity, including DoS attacks, scanning, worms, unexpected application services, and policy violations, such as a client system providing network services to other systems. 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 and attacks that have unusual flow patterns. Most NBA sensors can also reconstruct a series of observed events to determine the origin of a threat.

NBA products automatically update their baselines on an ongoing basis. As a result, typically there is not much tuning or customization to be done, other than updating firewall ruleset-like policies that most products support. A few NBA products offer limited signature customization capabilities; these are most helpful for inline sensors because they can use the signatures to find and block attacks that a firewall or router might not be capable of blocking. Besides reviewing tuning and customizations periodically to ensure that they are still accurate, administrators should also ensure that significant changes to hosts are incorporated, such as new hosts and new services. Generally it is not feasible to automatically link NBA systems with change management systems, but administrators could review change management records regularly and adjust host inventory information in the NBA to prevent false positives.

NBA technologies have some significant limitations. They are delayed in detecting attacks because of their data sources, especially when they rely on flow data from routers and other network devices. This data is often transferred to the NBA in batches from every minute to a few times an hour. Attacks that occur quickly 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; however, this is much more resource-intensive than analyzing flow data. Also, a single sensor can analyze flow data from many networks, while a single sensor can generally directly monitor only a few networks at once. Therefore, to do direct monitoring instead of using flow data, organizations might have to purchase more powerful sensors and/or more sensors.

7. 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 characteristics a host-based IDPS might monitor are wired and wireless network traffic (only for that host), system logs, running processes, file access and modification, and system and application configuration changes. This section provides a detailed discussion of host-based IDPS technologies. First, it covers the major components of the technologies and explains the architectures typically used for deploying the components. It also examines the security capabilities of the technologies in depth, including the methodologies they use to identify suspicious activity. The rest of the section discusses the management capabilities of the technologies, including recommendations for implementation and operation.

7.1 Components and Architecture

This section describes the major components of typical host-based IDPSs and illustrates the most common network architectures for these components. It also provides recommendations for selecting which hosts should use host-based IDPSs. This section also describes how host-based IDPSs can affect a host's internal architecture, such as intercepting process calls.

7.1.1 Typical Components

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 if IDPS capabilities are enabled, also performs prevention actions. Section 7.2.2 discusses the types of activity monitored by host-based IDPSs. The agents transmit data to management servers, which may optionally use database servers for storage.[42] Consoles are used for management and monitoring.

Some host-based IDPS products use dedicated appliances running agent software instead of installing agent software on individual hosts. Each appliance is positioned to monitor the network traffic going to and from a particular host. Technically, these appliances could be considered network-based IDPSs, because they are deployed inline to monitor network traffic. However, they usually monitor activity for only one specific type of application, such as a Web server or database server, so they are more specialized than a standard network-based IDPS. Also, the software running on the appliance often has the same or similar functionality as the host-based agents. Therefore, host-based IDPS products using appliance-based agents are included in this section.

Each agent is typically designed to protect one of the following:

- **A server.** Besides monitoring the server's operating system (OS), the agent may also monitor some common applications.
- **A client host (desktop or laptop).** Agents designed to monitor users' hosts usually monitor the OS and common client applications such as e-mail clients and Web browsers.
- **An application service.** Some agents perform monitoring for a specific application service only, such as a Web server program or a database server program. This type of agent is also known as an *application-based IDPS*.

[42] Because this publication focuses on enterprise IDPS deployment, it assumes that agents send their data to management servers; however, some agents can be deployed standalone, and managed and monitored directly by the host's administrators without using a management server.

Most products do not have agents for other types of hosts, such as network devices (e.g., firewalls, routers, switches).

7.1.2 Network Architectures

The network architecture for host-based IDPS deployments is typically very 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. Most products encrypt their communications, preventing eavesdroppers from accessing sensitive information. Appliance-based agents are typically deployed inline immediately in front of the hosts that they are protecting. Figure 7-1 shows an example of a host-based IDPS deployment architecture.

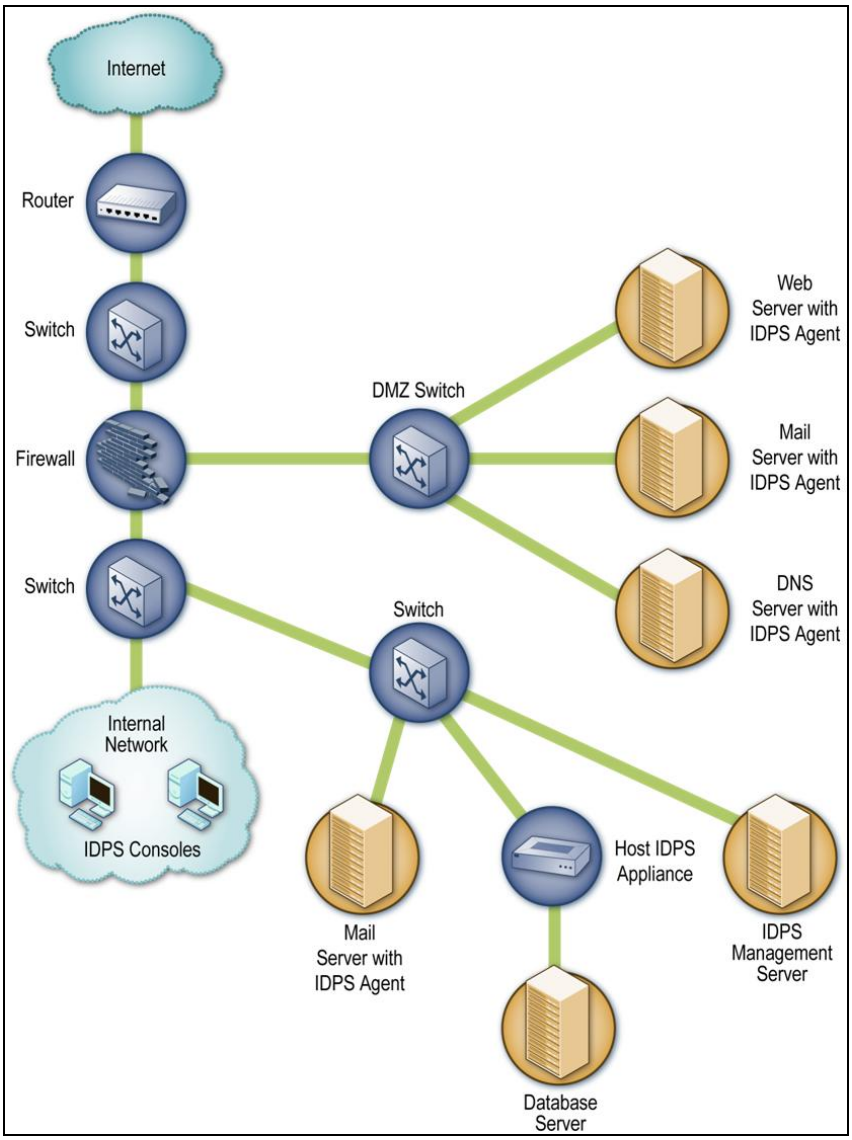

Figure 7-1. Host-Based IDPS Agent Deployment Architecture Example

7.1.3 Agent Locations

Host-based IDPS agents are most commonly deployed to critical hosts such as publicly accessible servers and servers containing sensitive information. However, because agents are available for various server and desktop/laptop operating systems, as well as specific server applications, organizations could potentially deploy agents to most of their servers and desktops/laptops. Some organizations use host-based IDPS 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. Organizations should consider the following additional criteria when selecting agent locations:

- The cost to deploy, maintain, and monitor the agents
- The OSs and applications supported by the agents
- The importance of the host's data or services
- The ability of the infrastructure to support the agents (e.g., sufficient network bandwidth to transfer alert data from the agents to centralized servers and to transfer software and policy updates from the centralized servers to the agents).

7.1.4 Host Architectures

To provide intrusion prevention capabilities, most IDPS agents alter the internal architecture of the hosts on which they are installed. 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., e-mail, Web).

Some host-based IDPS agents do not alter the host architecture. Instead, they monitor activity without shims, or they analyze the artifacts of activity, such as log entries and file modifications. Although less intrusive to the host, reducing the possibility of the IDPS interfering with the host's normal operations, these methods are also generally less effective at detecting threats and often cannot perform any prevention actions.

One of the important decisions in selecting a host-based IDPS solution is whether to install agents on hosts or use agent-based appliances. From a detection and prevention perspective, installing agents on hosts is generally preferable because the agents have direct access to the hosts' characteristics, often allowing them to perform more comprehensive and accurate detection and prevention. However, agents often support only a few common OSs; if a host does not use a supported OS, an appliance can be deployed instead. Another reason to use an appliance instead of installing an agent on a host is performance; if an agent would negatively impact the performance of the monitored host too much, it might be necessary to offload the agent's functions to an appliance.

7.2 Security Capabilities

Host-based IDPSs provide a variety of security capabilities. Sections 7.2.1 through 7.2.4 describe common security capabilities, divided into four categories: logging, detection, prevention, and other, respectively.

7.2.1 Logging Capabilities

Host-based IDPSs typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, to investigate incidents, and to correlate events between the host-based IDPS and other logging sources. Data fields commonly logged by host-based IDPSs include the following:

- Timestamp (usually date and time)
- Event or alert type
- Rating (e.g., priority, severity, impact, confidence)
- Event details specific to the type of event, such as IP address and port information, application information, filenames and paths, and user IDs
- Prevention action performed (if any).

7.2.2 Detection Capabilities

Most host-based IDPSs have the capability to 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. This section discusses the following aspects of host-based IDPS detection capabilities:

- Types of events detected
- Detection accuracy
- Tuning and customization
- Technology limitations.

7.2.2.1 Types of Events Detected

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. For example, several products only analyze network traffic, and other products only check the integrity of a host's critical files. Specific techniques commonly used in host-based IDPSs include the following:

- **Code Analysis.** Agents might use one or more of the techniques listed below to identify malicious activity by analyzing attempts to execute code. All of these techniques are helpful at stopping malware and can also prevent other attacks, such as some that would permit unauthorized access, code execution, or escalation of privileges.
  - **Code behavior analysis.** Before code is run normally on a host, it can first be executed in a virtual environment or a sandbox to analyze its behavior and compare it to profiles or rules of known good and bad behavior. For example, when a particular piece of code is executed, it might attempt to gain administrator-level privileges or to overwrite a system executable.
  - **Buffer overflow detection**. Attempts to perform stack and heap buffer overflows can be detected by looking for their typical characteristics, such as certain sequences of instructions and attempts to access portions of memory other than those allocated to the process.

  - **System call monitoring**. The agent knows which applications and processes should be calling which other applications and processes or performing certain actions. For example, an agent could recognize a process attempting to intercept keystrokes, such as a keylogger. Another example is an agent that restricts component object model (COM) object loading, such as permitting a PDA application, but not other applications, to access an e-mail client's address book. Agents can also restrict which drivers can be loaded, which can prevent the installation of rootkits and other attacks.

  - **Application and library lists.** An agent might monitor each application and library (e.g., dynamic link library [DLL]) that a user or process attempts to load and compare that information to lists of authorized and unauthorized applications and libraries. This can be used not only to restrict which applications and libraries can be used, but which versions of them can be used.

- **Network Traffic Analysis.** This is often similar to what a network-based IDPS does; some products can analyze both wired and wireless network traffic. In addition to network, transport, and application layer protocol analysis, agents may include special processing for common applications, such as popular e-mail clients. Traffic analysis also allows the agent to extract files sent by applications such as e-mail, 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). Some of these firewalls can generate and use a list of the hosts with which this host should be communicating, particularly within the organization.

- **Filesystem Monitoring.** Filesystem monitoring can be performed using several different techniques, including the ones listed below. Administrators should be aware that some products base their monitoring on filenames, so if users or attackers alter filenames, filesystem monitoring techniques might be made ineffective.

  - **File integrity checking**. This involves periodically generating message digests or other cryptographic checksums for critical files, comparing them to reference values, and identifying differences. File integrity checking can only determine after-the-fact that a file has already been changed, such as a system binary being replaced by a Trojan horse or a rootkit.

  - **File attribute checking.** This is periodically checking the attributes of important files, such as ownership and permissions, for changes. Like file integrity checking, it can only determine after-the-fact that a change has occurred.

  - **File access attempts.** An agent with a filesystem shim can monitor all attempts to access critical files, such as system binaries, and stop attempts that are suspicious. The agent has a set of policies regarding file access, so the agent compares those policies to the characteristics of the current attempt, including which user or application is trying to access each file, and what type of access has been requested (read, write, execute).[43] This could be used to prevent some forms of malware from being installed, such as rootkits and Trojan horses, as well as preventing many other types of malicious activity involving file access, modification, replacement, or deletion.

[43] On Windows systems, many configuration settings reside in a set of special files known as the *registry*. Some agents have special registry shims that restrict access to critical portions of the registry, especially those frequently used by malware.

- **Log Analysis.** Some agents can monitor and analyze OS and application logs to identify malicious activity.[44] These logs may contain information on system events, which are operational actions performed by OS components (e.g., shutting down the system, starting a service); audit records, which contain security event information such as successful and failed authentication attempts and security policy changes; and application events, which are significant operational actions performed by applications, such as application startup and shutdown, application failures, and major application configuration changes.
- **Network Configuration Monitoring.** Some agents can monitor a host's current network configuration and detect changes to it. Typically all network interfaces on the host are monitored, including wired, wireless, virtual private network (VPN), and modem. Examples of significant network configuration changes are network interfaces being placed in promiscuous mode, additional TCP or UDP ports being used on the host, or additional network protocols being used, such as non-IP protocols. These changes could indicate that the host has already been compromised and is being configured for use in future attacks or for transferring data.

Organizations should determine which aspects of hosts need to be monitored and select IDPS products that provide adequate monitoring and analysis for them.

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

7.2.2.2 Detection Accuracy

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 several of the possible detection techniques, such as log analysis and filesystem monitoring, do not have knowledge of the context under which detected events occurred. For example, a host may be rebooted, a new application installed, or a system file replaced. These actions could be done by malicious activity, or they could be part of normal host operation and maintenance. The events themselves are detected accurately, but their benign or malicious nature cannot always be determined without additional context. Some products, particularly those intended for desktop/laptop use, prompt users to provide context, such as whether or not the user is currently upgrading a particular application. If a user does not respond to the prompt in a set period of time (typically a few minutes), the agent chooses a default action (allow or deny).

Host-based IDPSs that use combinations of several detection techniques should generally be capable of achieving 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 collect more information on the activities occurring. This provides a more complete picture of the events, and may also provide additional context that can be helpful in assessing the intent of certain events.

7.2.2.3 Tuning and Customization

Host-based IDPSs usually require considerable tuning and customization. For example, many rely on observing host activity and developing baselines or profiles of expected behavior. Others need to be

[44] Some products only perform log analysis and log management activities, such as log consolidation. Although these products are often referred to as host-based IPS, some of them are actually security information and event management (SIEM) products. Section 9 contains additional information on SIEM.

configured with detailed policies that define exactly how each application on a host should behave. As the host environment changes, administrators should ensure that host-based IDPS policies are updated to take those changes into account. Generally it is not feasible to automatically link host-based IDPSs with change management systems, but administrators could review change management records regularly and adjust host configuration and policy information in the host-based IDPS to prevent false positives.

Policies can often be set on a per-host basis or for groups of hosts, which provides flexibility. Some products also 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 IDPSs also offer whitelist and blacklist capabilities for hosts (e.g., IP addresses of other hosts with which a host might communicate), applications, ports, filenames, and other host characteristics. In fact, some products automatically update agents with the latest whitelist and blacklist information, based on reports from other agents of newly detected malicious activity. Another common feature of host-based IDPSs is customizing each alert, such as specifying which response option should be performed for an alert.

The sophistication of signature capabilities for host-based IDPSs varies widely depending on the detection techniques used by each product.

7.2.2.4 Technology Limitations

Host-based IDPSs have some significant limitations. Some of these limitations are described in Section 7.2.2.2. Other important limitations include the following:

- **Alert Generation Delays.** 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.
- **Centralized Reporting Delays.** Many host-based IDPSs are intended to forward their alert data to the management servers on a periodic basis, not in a near-real-time fashion. Alert data is typically transferred in batches every 15 to 60 minutes to reduce overhead for the IDPS components and the network. Smaller host-based IDPS implementations can usually transfer data more often, but for larger implementations, vendors typically recommend less frequent transfers. This can cause delays in initiating response actions, which especially increases the impact of incidents that spread quickly, such as malware infestations.
- **Host Resource Usage.** Unlike the other IDPS technologies, host-based IDPSs involve running agents on the hosts being monitored. These agents can consume considerable host resources, including memory, processor usage, and disk storage. The agents' operation, particularly the shims, can also cause slowdowns in operations such as network and filesystem usage. Testing of host resource usage should be performed when evaluating host-based IDPS products for possible purchase.
- **Conflicts with Existing Security Controls.** Installing an agent can cause existing host security controls to be disabled automatically, such as personal firewalls, if those controls are perceived to duplicate functionality provided by the agent. Installing an agent can also cause conflicts with other security controls, especially those that use shims to intercept host activity (e.g., personal firewalls, VPN clients). For some products, a network shim is optional, although it does permit greater functionality, especially in prevention actions. To identify any potential conflicts, implementers should test agents on hosts that are running the host security controls used on the hosts to which the agents would be deployed.

- **Rebooting Hosts.** For many host-based IDPS products, agent software upgrades and some agent configuration changes can necessitate rebooting the monitored hosts. As with other problems mentioned earlier, implementers should perform extensive testing of this before selecting products and consider the impact that reboots could have on the effectiveness of the agents (e.g., agents being unable to detect the latest threats because important hosts could not be rebooted).

7.2.3 Prevention Capabilities

Host-based IDPS agents offer various intrusion prevention capabilities. Because the capabilities vary based on the detection techniques used by each product, the following items describe the capabilities by detection technique.

- **Code Analysis.** The code analysis techniques can prevent code from being executed, including malware and unauthorized applications. Some host-based IDPSs can also stop network applications from invoking shells, which could be used to attempt to perform certain types of attacks. If configured and tuned well, code analysis can be very effective, particularly at stopping previously unknown attacks.
- **Network Traffic Analysis.** This can stop incoming network traffic from being processed by the host and outgoing network traffic from exiting it. This might be done to stop network, transport, and application layer attacks (and in some cases, wireless networking protocol attacks), as well as to stop the use of unauthorized applications and protocols. Analysis can also identify malicious files being downloaded or transferred and prevent those files from being placed on the host. The network traffic might be dropped or rejected, and the host's personal firewall (which might be built into the agent) could be reconfigured to shun additional traffic related to the suspicious traffic. Network traffic analysis is effective at stopping many known and previously unknown attacks.
- **Network Traffic Filtering.** Working as a host-based firewall, this can stop unauthorized access and acceptable use policy violations (e.g., use of inappropriate external services). It is effective only against stopping activity that is identifiable by IP address and TCP port, UDP port, or ICMP type and code.
- **Filesystem Monitoring.** This 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. This technique can provide an additional layer of access control to supplement the existing access control technologies on a host.

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 well after they have occurred.

7.2.4 Other Capabilities

Some host-based IDPSs offer non-IDPS capabilities such as antivirus software, spam filtering, and Web or e-mail content filtering. It is outside the scope of this guide to discuss these capabilities, which are often provided by bundling separate products with the IDPS software. This section focuses on those additional product capabilities that are more closely tied to host-based IDPS functionality. Examples of these capabilities are as follows:

- **Removable Media Restriction.** Some products can enforce restrictions on the use of removable media, both USB-based (e.g., flash drive) and traditional (e.g., CD, floppy disk). This can prevent

malware or other unwanted files from being transferred to a host, and can also stop sensitive files from being copied from the host to removable media.

- **Audiovisual Device Monitoring**. A few host-based IDPS products can detect when a host's audiovisual devices, such as microphones, cameras, or IP-based phones are activated or used. This could indicate that the host has been compromised by an attacker.
- **Host Hardening**. Some host-based IDPSs can automatically harden hosts on an ongoing basis. For example, if an application is reconfigured, causing a particular security function to be disabled, the IDPS could detect this and enable the security function.
- **Process Status Monitoring**. Some products monitor the status of processes or services running on a host, and if they detect that one has stopped, they restart it automatically. Some products can also monitor the status of security programs such as antivirus software.
- **Network Traffic Sanitization**. Some agents, particularly those deployed on appliances, can sanitize the network traffic that they monitor. For example, an appliance-based agent could act as a proxy and rebuild each request and response that is directed through it. This can be effective at neutralizing certain unusual activity, particularly in packet headers and application protocol headers. Sanitization performed by an appliance can also reduce the amount of reconnaissance the attackers can perform on the host it is protecting. Examples include hiding the servers' OS fingerprints and application error messages. Some products can also prevent sensitive information such as social security numbers and credit card numbers from being displayed on Web server pages.

7.3 Management

Most host-based IDPSs offer similar management capabilities. This section discusses major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently.

7.3.1 Implementation

Once a host-based IDPS product has been selected, the administrators need to design an architecture, perform IDPS component testing, secure the IDPS components, and then deploy them. The following items list additions to the material presented in Section 3.3.1:

- **Component Testing and Deployment.** After the host-based IDPS components have been evaluated in a test environment, organizations should implement a small pilot in the production environment. This allows administrators to perform tuning and customization activities on a small set of production hosts in preparation for a larger deployment. The prevention features should be disabled during the pilot and the subsequent production implementation until the agents have been sufficiently tuned and customized.
- **Securing the Components.** If the management servers or consoles must authenticate to each agent host to manage the agents or collect their data, organizations should ensure that the authentication mechanisms can be managed and secured properly. For example, if passwords are needed, there are security concerns with using a single password for all agent hosts; if a separate password is used for each agent host, the passwords can be difficult to track and maintain for hundreds or thousands of agents. If cryptographic keys are used for authentication, key management can present challenges in issuing and distributing keys.

7.3.2 Operation

Host-based IDPSs should be operated according to the recommendations presented in Section 3.3.2. The only exception is in updating the agents. Some agents can periodically check the management server for updates and automatically retrieve and install or apply those updates. Other agents cannot do this, requiring an administrator to manually check for, transfer, and install or apply updates. In many cases, an agent's update capability is related to the type of operating system on which it is deployed.

7.4 Summary

Host-based IDPSs 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 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 if prevention capabilities are enabled, also performs prevention actions. The agents transmit data to management servers. Each agent is typically designed to protect a server, a desktop or laptop, or an application service.

The network architecture for host-based IDPS deployments is typically very 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 management network. Host-based IDPS agents are most commonly deployed to critical hosts such as publicly accessible servers and servers containing sensitive information. However, because agents are available for various server and desktop/laptop operating systems, as well as specific server applications, organizations could potentially deploy agents to most of their servers and desktops/laptops. Organizations should consider several criteria when selecting agent locations, including the need to analyze activity that cannot be monitored by other security controls; the cost of the agents' deployment, maintenance, and monitoring; the OSs and applications supported by the agents; the importance of each host's data or services; and the ability of the network infrastructure to support the agents' communications.

Most IDPS agents alter the internal architecture of the hosts on which they are installed through shims, which are layers of code placed between existing layers of code. Although it is less intrusive to the host to perform monitoring without shims, which reduces the possibility of the IDPS interfering with the host's normal operations, monitoring without shims is also generally less accurate at detecting threats and often precludes the performance of effective prevention actions.

Host-based IDPSs provide a variety of security capabilities. They typically perform extensive logging of data related to detected events and can detect several types of malicious activity. Detection techniques used include code analysis, network traffic analysis, network traffic filtering, filesystem monitoring, log analysis, and network configuration monitoring. Host-based IDPSs that use combinations of several detection techniques should generally be capable of achieving more accurate detection than products that use one or a few techniques, because each technique can monitor different characteristics of hosts. Organizations should determine which characteristics need to be monitored and select IDPS products that provide adequate monitoring and analysis of those characteristics.

Host-based IDPSs usually require considerable tuning and customization. For example, many rely on observing host activity and developing baselines or 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, administrators should ensure that host-based IDPS policies are updated to take those changes into account.

Host-based IDPSs have some significant limitations. Some detection techniques are performed only periodically, such as hourly or a few times a day, to identify events that have already happened, causing significant delay in identifying certain events. Also, many host-based IDPSs forward their alert data to the management servers in batches a few times an hour, which can cause delays in initiating response actions. Because host-based IDPSs run agents on the hosts being monitored, they can impact host performance because of the resources the agents consume. Installing an agent can also cause conflicts with existing host security controls, such as personal firewalls and VPN clients. Agent upgrades and some configuration changes can also necessitate rebooting the monitored hosts.

Host-based IDPSs offer various intrusion prevention capabilities; these vary based on the detection techniques used by each product. Code analysis techniques can prevent code from being executed; this can be very effective at stopping both known and previously unknown attacks. Network traffic analysis can stop incoming and outgoing network traffic containing network, transport, or application layer attacks, wireless networking protocol attacks, and the use of unauthorized applications and protocols. Network traffic filtering works as a host-based firewall and stops unauthorized access and acceptable use policy violations. Filesystem monitoring can prevent files from being accessed, modified, replaced, or deleted, which can stop malware installation and other attacks involving inappropriate file access. Other host-based IDPS detection techniques generally do not support prevention actions because they identify events well after they have occurred.

Some host-based IDPSs offer additional capabilities related to intrusion detection and prevention, such as enforcing restrictions on the use of removable media, detecting the activation or use of audiovisual devices, automatically hardening hosts on an ongoing basis, monitoring the status of running processes and restarting failed ones, and performing network traffic sanitization.

This page has been left blank intentionally.

8. Using and Integrating Multiple IDPS Technologies

As Sections 4 through 7 have explained, the four primary types of IDPS technologies—network-based, wireless, network behavior analysis (NBA), and host-based—each offer fundamentally different information gathering, logging, detection, and prevention capabilities. Each technology type offers benefits over the other, such as detecting some events that the others cannot, detecting some events with significantly greater accuracy than the other technologies, and performing in-depth analysis without significantly impacting the performance of the protected hosts. Accordingly, organizations should consider using multiple types of IDPS technologies to achieve more comprehensive and accurate detection and prevention of malicious activity, with lower rates of false positives and false negatives. This section provides guidance on using multiple IDPS technologies to create a broader IDPS solution and discusses the advantages and disadvantages of using multiple technologies.

Organizations that are planning to use multiple types of IDPS technologies, or even multiple products within a single IDPS technology class, should consider whether or not the IDPS products should be integrated in some way, either working together directly or feeding their data into a centralized logging system or security information and event management system. This section explains how different IDPS products can be integrated, and the benefits and limitations of the integration methods. It also provides overviews of other technologies that complement IDPS technologies and discusses how they can be included in an IDPS solution to further improve detection and prevention.

8.1 The Need for Multiple IDPS Technologies

In many environments, a robust IDPS solution cannot be achieved without using multiple types of IDPS technologies. For example, network-based IDPSs cannot monitor wireless protocols, and wireless IDPSs cannot monitor application protocol activity. Table 8-1 provides a high-level comparison of the four primary IDPS technology types. The strengths listed in the table indicate the roles or situations in which each technology type is generally superior to the others. A particular technology type may have additional benefits over others, such as logging additional data that would be useful for validating alerts recorded by other IDPSs, or preventing intrusions that other IDPSs cannot because of technology capabilities or placement (e.g., on the host instead of on the network).

Table 8-1. Comparison of IDPS Technology Types

IDPS Technology Type	Types of Malicious Activity Detected	Scope per Sensor or Agent	Strengths
Network-Based	Network, transport, and application TCP/IP layer activity	Multiple network subnets and groups of hosts	Able to analyze the widest range of application protocols; only IDPS that can thoroughly analyze many of them
Wireless	Wireless protocol activity; unauthorized wireless local area networks (WLAN) in use	Multiple WLANs and groups of wireless clients	Only IDPS that can monitor wireless protocol activity
NBA	Network, transport, and application TCP/IP layer activity that causes anomalous network flows	Multiple network subnets and groups of hosts	Typically more effective than the others at identifying reconnaissance scanning and DoS attacks, and at reconstructing major malware infections
Host-Based	Host application and operating system (OS) activity; network, transport, and application TCP/IP layer activity	Individual host	Only IDPS that can analyze activity that was transferred in end-to-end encrypted communications

For most environments, a combination of network-based and host-based IDPSs is needed for an effective IDPS solution. Wireless IDPSs may also be needed if the organization determines that its wireless networks need additional monitoring or if the organization wants to ensure that rogue wireless networks are not in use in the organization's facilities. NBA products can also be deployed if organizations desire additional detection capabilities for denial of service (DoS) attacks, worms, and other threats as discussed in Section 6.

In addition to using multiple types of IDPS technologies, some organizations also use multiple products of the same IDPS technology type. This is often done to improve detection capabilities. Because each product uses somewhat different detection methodologies and detects some events that another product cannot, using multiple products can allow for more comprehensive detection of possible incidents. Also, having multiple products in use, particularly to monitor the same activity, makes it easier for analysts to confirm the validity of alerts and identify false positives, and also provides redundancy, should one product fail for any reason.

8.2 Integrating Different IDPS Technologies

Many organizations use multiple IDPS products, usually from different vendors (most vendors make products in only one IDPS technology type). By default, these products function completely independently of each other. This has some notable 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 IDPS users and administrators may have to expend extra effort to monitor and manage multiple sets of products. IDPS products can be directly integrated, such as one product feeding alert data to another product, or they can be indirectly integrated, such as all the IDPS products feeding alert data into a security information and event management system. Sections 8.2.1 and 8.2.2 discuss the benefits and limitations of direct and indirect integration, respectively.

8.2.1 Direct IDPS Integration

Direct IDPS integration is most often performed when an organization uses multiple IDPS products from a single vendor. For example, some vendors offer both network-based and host-based products. These vendors frequently offer a single console that can be used to manage and monitor both types of products. This can provide significant time savings to administrators and users because it streamlines their work. Some products also share data; for example, a product might use host-based IDPS data to determine if an attack detected by network-based IDPS sensors was successful, or if an attack stopped by network-based IDPS data would have been successful if allowed to pass. This information can speed the analysis process and help users to better 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 the integrated solution.

A more limited form of direct IDPS integration is having one IDPS product provide data for another IDPS product. As mentioned previously, two products from the same vendor often share data with each other for correlation purposes. Data can also be shared among products from different vendors, although typically this simply involves one product providing data as input to the second product. For example, a network-based IDPS could potentially provide network flow information to an NBA sensor. A host-based IDPS could provide system configuration information to NBA or network-based IDPS sensors. This data can be used for event correlation and better prioritization of alerts.

8.2.2 Indirect IDPS Integration

Indirect IDPS integration is usually performed with security information and event management (SIEM) software.[45] SIEM software is designed to import information from various security-related logs and correlate events among them.[46] 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, e-mail 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.[47] 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 products usually do not generate original event data; instead, they generate meta-events based on their analysis of the imported event data.

Ways in which SIEM software complements IDPSs include the following:

- SIEM software can identify some types of events that individual IDPSs cannot because of its ability to correlate events logged by different technologies.
- The consoles for SIEM software can make data from many sources available through a single interface, which can save time for users that need to monitor multiple IDPSs. SIEM consoles also may offer analysis and reporting tools that certain IDPSs' consoles do not.
- Users can more easily verify the accuracy of IDPS alerts because the SIEM may be able to link each alert to supporting information from other logs. This can also help users to determine whether or not certain attacks succeeded.

Limitations of SIEM software in the context of IDPS include the following:

- There is often a considerable delay between the time an event begins and the time the SIEM sees the corresponding log data. Log data may be transferred from logging hosts to the SIEM in batch mode, such as every 5 or 10 minutes. As a result, malicious activity alerts are often displayed on an IDPS console earlier than on a SIEM console, and prevention actions are less timely.
- SIEM products typically transfer only some data fields from the original logs. For example, if a network-based IDPS records packets, the packets may not be transferred to the SIEM because of bandwidth and storage limitations. Also, the log normalization process that converts each data field to a standard format and labels the data consistently can occasionally introduce errors in the data or cause some data to be lost. Fortunately, SIEM products typically do not alter the original data sources, so they can be referenced to verify the accuracy of the data if needed.
- SIEM software may not offer agents for all IDPS products. This could require administrators to write custom agents to transfer IDPS data to the SIEM servers, or it could necessitate having the IDPSs perform logging using a different mechanism so that the SIEM software can understand the log format.

---

[45] For additional information on SIEM software and log management, see NIST SP 800-92, *Guide to Computer Security Log Management*, which is available at http://csrc.nist.gov/publications/nistpubs/.

[46] SIEM is also sometimes known as security event management (SEM) or security information management (SIM).

[47] There are no widely accepted standards for IPS log formats or data fields. As a result, each IPS product uses its own schema for logging.

An alternative to using SIEM software for centralized logging is to use a solution based primarily on the syslog protocol.[48] 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. Generally, the use of syslog for centralized collection and analysis of IDPS logs does not provide sufficiently strong analysis capabilities to support incident identification and handling.

8.3 Other Technologies with IDPS Capabilities

In addition to dedicated IDPS technologies, organizations typically have several other types of technologies that offer some IDPS capabilities and complement the primary IDPSs. This section discusses common types of complementary technologies: network forensic analysis tools, anti-malware technologies (antivirus software and antispyware software), firewalls and routers, and honeypots.[49] For each, a brief overview of the technology is provided, and its use in intrusion detection and prevention and its relationship to IDPSs are explained. Recommendations are also made as applicable for how the complementary technologies should be used alongside of IDPSs.

8.3.1 Network Forensic Analysis Tool (NFAT) Software

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. In addition to its forensic capabilities, NFAT software also offers features that facilitate network traffic analysis, such as the following:

- Reconstructing events by replaying network traffic within the tool, ranging from an individual session (e.g., instant messaging [IM] between two users) to all sessions during a particular time period. The speed of the replaying can typically be adjusted as needed.
- Visualizing the traffic flows and the relationships among hosts. Some tools can even tie IP addresses, domain names, or other data to physical locations and produce a geographic map of the activity.
- Building profiles of typical activity and identifying significant deviations.
- Searching application content for keywords (e.g., “confidential”, “proprietary”).

This makes it more valuable for network forensics and less valuable for intrusion detection and prevention than a typical network-based IDPS.

---

[48] Although syslog has been in use for many years, it has not been standardized formally. Request for Comments (RFC) 3164, *The BSD Syslog Protocol*, was published in August 2001, and it is an informational RFC that describes commonly used syslog message formats based on existing implementations. It is available at http://www.ietf.org/rfc/rfc3164.txt. By default, syslog’s transport mechanism is trivially simple; RFC 3164 states that “…the payload of any IP packet destined to UDP port 514 MUST be considered to be a valid syslog message”. RFC 3195, *Reliable Delivery for Syslog*, was published in November 2001, and it defines multiple transport mechanisms for syslog. It is available at http://www.ietf.org/rfc/rfc3195.txt.

[49] Additional information on complementary tools is available from NIST SP 800-86, *Guide to Integrating Forensic Techniques into Incident Response*, which is available at http://csrc.nist.gov/publications/nistpubs/.

Ways in which NFAT software complements IDPSs include the following:

- NFAT software is often more valuable for network forensics than IDPS software because of its extensive packet logging.
- Having NFAT software perform packet logging can reduce the load on network-based IDPS sensors.
- NFAT software might be better-suited to customization, especially for content searches (e.g., keywords), than some IDPS technologies.
- Some NFAT graphical user interfaces (GUI) may offer analysis, visualization, and reporting capabilities that IDPS consoles do not.

Limitations of NFAT software in the context of IDPS include the following:

- NFAT software usually does not have the intrusion detection capabilities of network-based IDPSs.
- NFAT software typically offers no intrusion prevention capabilities.

8.3.2 Anti-Malware Technologies

The most commonly used technical control for malware threat mitigation is 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. Most organizations deploy antivirus software both centrally (e.g., e-mail servers, firewalls) and locally (e.g., file servers, desktops, laptops) so that all major malware entry vectors can be monitored.

Another commonly used control for malware threat mitigation is spyware detection and removal utilities, also known as antispyware software. They are similar to antivirus software, but they focus on detecting 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, popup ads, and Web browser hijacking.

Both antivirus and antispyware products detect threats primarily through signature-based analysis. To identify previously unknown threats, they also use heuristic techniques that examine activity for certain suspicious characteristics. The product vendors create and release additional signatures when new threats emerge, so that the products can detect them.

Ways in which antivirus and antispyware software complements IDPSs include the following:

- IDPSs usually have limited malware and spyware detection capabilities (often only for the most common threats, such as widespread worms), so antivirus and antispyware software can detect many threats that IDPSs cannot.
- NBA technology might identify that a worm is spreading based on unusual traffic flows, but it probably could not identify which worm it is. Antivirus software should be able to determine which worm it is, if the threat is not a new one for which the antivirus software does not yet have signatures.
- Antivirus software, and to a lesser extent antispyware software, can take some load from IDPSs, such as having antivirus software identify instances of a particular worm and disabling the worm's signatures on the IDPS sensors. This is particularly important during a widespread malware

infection, when IDPSs might become overwhelmed with worm alerts and other important events occurring at the same time might go unnoticed by IDPS users.

Limitations of antivirus and antispyware software in the context of IDPS include the following:

- Antivirus and antispyware software cannot detect threats other than malware and spyware.
- Network-based IDPS and NBA software are often better able to recognize network service worms than antivirus software can because antivirus software often monitors only the most common application protocols. Also, antispyware software typically cannot detect network service worms. Network-based IDPS and NBA software can typically monitor any protocol.
- For a new threat, antivirus and antispyware software often cannot recognize it until the vendor releases new signatures and updates are installed. In some cases, especially for threats with easily identifiable characteristics, an IDPS can detect the new threat during this window of time because IDPS administrators can write a custom signature for the IDPS. Antivirus and antispyware software typically do not permit administrators to write signatures. Also, NBA software can often recognize new worms by their anomalous traffic patterns.

8.3.3 Firewalls and Routers

Firewalls (network-based and host-based) and routers filter network traffic based on TCP/IP characteristics such as the source and destination IP addresses, the transport layer protocol (e.g., TCP, UDP, ICMP), and basic protocol information (e.g., TCP or UDP port numbers, ICMP type and code). Most firewalls and routers log which connections or connection attempts they block; the blocked activity is often generated by unauthorized access attempts from automated attack tools, port scanning, and malware. Some network-based firewalls also act as proxies. When a proxy is used, each successful connection attempt actually results in the creation of two separate connections: one between the client and the proxy server, and another between the proxy server and the true destination. Many proxies are application-specific, and some actually perform some analysis and validation of common application protocols, such as HTTP. The proxy may reject client requests that appear to be invalid (which could include some forms of attacks) and log information regarding these requests.

Ways in which firewalls and routers complement IDPSs include the following:[50]

- Network-based firewalls and routers often perform network address translation (NAT), which is the process of mapping addresses on one network to addresses on another network. NAT is most often accomplished by mapping private addresses from an internal network to one or more public addresses on a network that is connected to the Internet. Firewalls and routers that perform NAT typically record each NAT address and mapping. IDPS users may need to make use of this mapping information to identify the actual IP address of a host behind a device performing NAT.
- If IDPSs and other security controls (e.g., antivirus software) cannot stop a new network-borne threat, such as a network service worm or denial of service attack, firewalls or routers might have to be temporarily reconfigured to block the threat.
- As mentioned in Sections 4 through 7, many IDPSs can reconfigure firewalls or routers to block particular threats.
- Routers are often used as data sources for NBA deployments.

[50] Some firewalls and routers can run IDPS software. The discussion in this section addresses only the core capabilities of firewalls and routers, not add-on IDPS capabilities.

Limitations of firewalls and routers in the context of IDPS include the following:

- Firewalls and routers cannot detect most types of malicious activity.
- Firewalls and routers typically log relatively little information, such as the basic characteristics of denied connection attempts only, and they rarely record the content of any packets. NBA technologies and some network-based IDPSs can log much more information about network traffic than firewalls and routers do.

8.3.4 Honeypots

Some organizations are sufficiently concerned with detecting the earliest signs of widespread incidents, such as major new worms, that they deploy deceptive measures such as honeypots so that they can collect better data on these threats. *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.

8.4 Summary

The four primary types of IDPS technologies—network-based, wireless, NBA, and host-based—each offer fundamentally different information gathering, logging, detection, and prevention capabilities. Each technology type offers benefits over the other, such as detecting some events that the others cannot and detecting some events with significantly greater accuracy than the other technologies. Accordingly, organizations should consider using multiple types of IDPS technologies to achieve more comprehensive and accurate detection and prevention of malicious activity. In many environments, a robust IDPS solution cannot be achieved without using multiple types of IDPS technologies. For most environments, a combination of network-based and host-based IDPSs is needed for an effective IDPS solution. Wireless IDPSs may also be needed if the organization determines that its wireless networks need additional monitoring or if the organization wants to ensure that rogue wireless networks are not in use in the organization's facilities. NBA technologies can also be deployed if organizations desire additional detection capabilities for DoS attacks, worms, and other threats that NBAs are particularly good at detecting.

Organizations that are planning to use multiple types of IDPS technologies, or even multiple products within a single IDPS technology class, should consider whether or not the IDPS products should be integrated in some way. Direct IDPS integration is most often performed when an organization uses multiple IDPS products from a single vendor, by having a single console that can be used to manage and monitor the multiple products. Some products can also share data, which can speed the analysis process and help users to better prioritize threats. A more limited form of direct IDPS integration is having one IDPS product provide data for another IDPS product, such as a network-based IDPS providing network flow information to an NBA sensor.

Indirect IDPS integration is usually performed with security information and event management (SIEM) software, which is designed to import information from various security-related logs and correlate events among them. SIEM software complements IDPSs in several ways, including correlating events logged by different technologies, displaying data from many event sources, and providing supporting information from other sources to help users verify the accuracy of IDPS alerts. An alternative to using SIEM

software for centralized logging is the syslog protocol, which provides a simple standard framework for log generation, storage, and transfer that any IDPS can use if designed to do so. 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 message content, 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. Generally, the use of syslog for centralized collection and analysis of IDPS logs does not provide sufficiently strong analysis capabilities to support incident identification and handling.

In addition to dedicated IDPSs, organizations typically have several other types of technologies that offer some IDPS capabilities and complement, but do not replace, the primary IDPSs. These include network forensic analysis tools, anti-malware technologies (antivirus software and antispyware software), and firewalls and routers.

9. IDPS Product Selection

This section provides guidance on selecting IDPS products. It first discusses the identification of general requirements that the IDPS products should meet. Next, it provides sets of criteria that can be used to evaluate four aspects of IDPS technologies: security capabilities, performance, management, and life cycle cost. Finally, the section concludes with a brief discussion of performing hands-on and paper evaluations of products, and when each evaluation technique is most appropriate. This section assumes that an organization has already determined that a particular type of IDPS technology—network-based, wireless, network behavior analysis (NBA), or host-based—is needed. A comparison of the technology types, which can be helpful for selecting the one most appropriate for a particular need, is provided in Section 8.

Organizations should use risk management techniques to identify the security controls necessary to mitigate risk to an acceptable level. Although it may be tempting to simply choose a product, using a risk management process to choose the most effective blend of controls enhances an organization's security posture. An explanation of the risk management process is outside the scope of this document; NIST SP 800-30, *Risk Management Guide for Information Technology Systems*, contains additional information on it.

9.1 General Requirements

Before evaluating IDPS products, organizations should first define the general requirements that the IDPS solution and products should meet. The features provided by IDPS products and the methodologies that they use vary considerably, so a product that best meets one organization's requirements might not be suitable for meeting another organization's requirements. Also, a single IDPS product might not be able to meet all of an organization's requirements for a particular type of IDPS technology (e.g., network-based), necessitating the use of multiple IDPS products of the same technology type. This is most common for large environments and for environments in which IDPS technologies serve multiple operational purposes.

9.1.1 System and Network Environments

Evaluators first need to understand the characteristics of the organization's system and network environments, so that an IDPS can be selected that will be compatible with them and able to monitor the events of interest on the systems and/or networks. This knowledge is also needed to design the IDPS solution and determine how many components (e.g., sensors, agents) will be needed and where they will be deployed (e.g., which systems will run IDPS agents, which network segments will be monitored). Characteristics to consider include the following:

- **Technical specifications of the IT environment.** Examples are as follows:
  - Network diagrams and maps specifying the architecture (both logical and geographical) of the network, including all connections to other networks, and the number and locations of hosts
  - The operating systems (OS), network services, and applications run by each host that might need to be protected by the IDPS[51]
  - The attributes of non-security systems with which the IDPS might need to be integrated, such as network management systems.

[51] In some cases, particularly for some host-based IDPSs, it may also be necessary to identify the application versions that need to be protected, so that it can be confirmed that the IDPS provides support for those versions.

- **Technical specifications of the existing security protections.** Examples of relevant protections are as follows:
  - Existing IDPS implementations
  - Centralized logging servers and SIEM software
  - Antimalware software, such as antivirus and antispyware software
  - Content filtering software, including antispam software
  - Network firewalls, routers, proxies, and other packet filtering devices and software
  - Communication encryption services, including link encryptors, virtual private networks (VPN), and Secure Sockets Layer (SSL)/Transport Layer Security (TLS).

9.1.2 Goals and Objectives

After gaining an understanding of the existing system and network environments, evaluators should articulate the technical, operational, and business goals and objectives they wish to attain by using an IDPS. The following questions should be considered in this area:

- **The types of threats for which the IDPS should provide protection.** Evaluators should state, as specifically as possible, the concerns that the organization has regarding the types of threats that originate both outside the organization and inside the organization (insider threats). Insider threats should encompass not only users who attack the system from within, but also authorized users who overstep their privileges, thereby violating organizational security policy or laws.
- **Any needs to monitor system and network usage for acceptable use violations or non-security reasons.** In some organizations, there are system use policies that target user behaviors that may be considered personnel management rather than system security issues. These might include accessing Web sites that provide content of questionable taste or value (such as pornography) or using the organization's systems to send email or other messages to harass individuals. Some IDPSs provide features that accommodate detecting such events. Monitoring usage can also assist organizations in determining when systems and networks are reaching capacity and might need to be upgraded or replaced.

9.1.3 Security and Other IT Policies

Evaluators should review their existing security policies and other IT policies before selecting products. The policies serve as a specification for many of the features that the IDPS products need to provide.[52] Examples of policy elements that can contain helpful information for IDPS product selection are as follows:

- **The goals of the policies.** It is helpful to articulate the goals outlined in the policies in terms of the standard security goals (integrity, confidentiality, and availability) as well as more generic management goals (privacy, protection from liability, manageability).
- **Reasonable use policies or other management provisions.** As mentioned above, many organizations have system use policies included as part of security policies and other IT policies.

[52] If the existing policies do not provide enough information on what types of activity should be permitted or denied, it may be necessary to revise the policies first before selecting an IDPS product to enforce the policies.

- **Processes for dealing with specific policy violations.** It is helpful to have a clear idea of what the organization wishes to do when an IDPS detects that a policy has been violated. If the organization does not intend to react to such violations, it may not make sense to configure the IDPS to detect them. If the organization wishes to respond to such violations, it may be necessary to select an IDPS product that can detect them, and perhaps also perform automated responses to stop them.

9.1.4 External Requirements

Evaluators should understand if the organization is subject to oversight or review by another organization, or if it is likely that the organization will be subject to an additional form of oversight in the near term. If either is true, the evaluators should determine if that oversight authority requires IDPSs or other specific security resources. Examples of external requirements are as follows:

- **Security-specific requirements levied by law.** For example, there may be legal requirements to protect personally identifiable information (such as earnings information or medical records) stored on the systems. There could also be legal requirements for investigation of security violations that divulge or endanger that information.
- **Audit requirements for security best practices or due diligence.** The audit requirements may specify functions that the IDPS must provide or support. Some IDPSs offer features to meet special needs of certain industries or market niches, such as reports designed to meet legislative requirements for health care or financial institutions.
- **System accreditation requirements.** If the organization's systems are subject to accreditation, the evaluators should identify and consider the accreditation authority's requirements for IDPS or other security protection.
- **Requirements for law enforcement investigation and resolution of security incidents.** They may impose additional requirements on IDPS functions, especially those having to do with collection and protection of IDPS logs as evidence.
- **Requirements to purchase products previously evaluated through an independent process.** For example, an organization might be required to or prefer to purchase products that hold a certain rating from an evaluating body.
- **Cryptography requirements.** For example, Federal agencies are required to purchase products that use FIPS-approved encryption algorithms to protect network communications and storage of sensitive data. Also, if any of the IDPS components will be deployed in other countries, evaluators should consider any restrictions or limitations that might affect the deployment or use of cryptographic components incorporated in the IDPS.

9.1.5 Resource Constraints

IDPSs can protect the systems of an organization, but at a price. It makes little sense to incur additional expense for IDPS features if the organization does not have sufficient systems or personnel to use them. Evaluators should consider the following:

- **The budget for acquisition and life cycle support of IDPS hardware, software, and infrastructure.** The total cost of ownership of IDPSs well exceeds acquisition costs. Other costs may be associated with acquiring systems on which to run software components, deploying additional networks, providing sufficient storage for IDPS data, obtaining specialized assistance in installing and configuring the system, and training personnel. See Section 9.5 for additional information on life cycle costs.

- **The staff needed to monitor and maintain an IDPS.** Some IDPSs are designed under the assumption that personnel will be available to monitor and maintain them around the clock. If evaluators do not anticipate having such personnel available, they may wish to explore those systems that accommodate less than full-time attendance or are designed for unattended use, or they could consider the possibility of outsourcing the monitoring and possibly also the maintenance of the IDPS.[53]

9.2 Security Capability Requirements

In addition to defining general requirements, as described in Section 9.1, evaluators also need to define more specialized sets of requirements. This section specifically addresses security capability requirements. Sections 9.3 through 9.5 discuss performance, management, and life cycle cost requirements, respectively. The criteria in these sections are presented as possible evaluation criteria and are not intended to be used as-is for performing product evaluations. Instead, organizations could use them as a basis for creating an organization-specific set of criteria that takes into account an organization's environment, policies, and existing security and network infrastructure. Section 9.6 provides additional information on performing IDPS evaluations.

Evaluating the security capabilities of each IDPS product is obviously very important. If the product cannot provide the necessary capabilities, then it ultimately is insufficient as a security control alone, and either a different product should be selected or the product should be used in conjunction with other security controls, such as a different IDPS product. This section presents IDPS security capability considerations in four categories: information gathering, logging, detection, and prevention. Section 9.6 provides guidance on collecting data on IDPS security capabilities as part of an evaluation.

9.2.1 Information Gathering Capabilities

Organizations should identify the information gathering capabilities needed for their IDPS's detection methodologies and analysis functions, and evaluate each IDPS product under consideration for its ability to offer those capabilities. Information on information gathering capabilities for each type of IDPS technology are presented in Sections 4 through 7.

9.2.2 Logging Capabilities

Organizations should carefully examine the event and alert logging capabilities of each IDPS solution being evaluated. The quality of logging, both completeness and precision, affects an organization's ability to perform analysis, confirm the accuracy of alerts, and correlate logged events with events recorded by other sources (e.g., other security controls, OS logs). IDPS products should log basic information at a minimum, such as a timestamp, the event type, the source of the event, and the sensor or agent that detected the event. Each IDPS product should also log supporting data involving the details of the event; these data fields are specific to particular IDPS product types, and common data fields are listed in Sections 4 through 7. IDPS products should also provide a mechanism that allows users to associate each log entry with corresponding external references, including Common Vulnerabilities and

[53] If portions of the IDPS will be or might be outsourced, organizations should ensure that their product requirements reflect outsourcing-specific requirements, such as limiting the actions that the outsourcers can perform and performing auditing of their actions.

Exposures (CVE) numbers,[54] which provide universal identifiers for vulnerabilities, and possibly other references such as vendor security advisories.

9.2.3 Detection Capabilities

Organizations should carefully evaluate the detection capabilities of each IDPS solution being evaluated. For many implementations, the detection capabilities are the most important function. Comparing detection capabilities is a complex undertaking because each product typically performs detection of a somewhat different set of events using different methodologies. Factors that organizations should consider in their IDPS evaluations include the following:

- Which types of activities it currently analyzes fully and analyzes partially, as well as future plans for additional analysis capabilities. Examples are as follows:
  - For network-based IDPS, a listing of the network, transport, and application layer protocols analyzed, and an explanation of the amount of analysis performed on each (e.g., signature-based detection, anomaly-based detection, stateful protocol analysis)
  - For host-based IDPS, a listing of the specific resources that can be monitored (e.g., log files, system files, network interfaces) and an explanation of how each is monitored (e.g., after-the-fact detection of changes, active handling of file access requests, TCP/IP stack monitoring)
- What types of incidents it can identify, such as denial of service [DoS] attacks, backdoors, policy violations, port scans, malware (e.g., worms, Trojan horses, rootkits, malicious mobile code), and unauthorized application/protocol use.
- How comprehensive its detection is for each type of incident it can identify (e.g., how many worms, how many types of DoS attacks).
- How effective its default, out-of-the-box configuration is. When an IDPS is first activated, its default settings should be reasonable. For example, signatures or policies that tend to generate large numbers of false positives should be disabled, and signatures or policies that are reliable and identify important recent attacks should be enabled. Detection thresholds (e.g., x instances in y minutes) should be set to values that attempt to balance false positives and false negatives. Also, features that are particularly resource-intensive should be disabled.
- How effective it is at detecting known malicious events, such as attacks, scans, or malware. Signature-based detection techniques typically perform better than anomaly detection and stateful protocol analysis techniques in recognizing known events. This should include the IDPS's ability to state precisely which exploit was performed and which vulnerability was targeted (e.g., CVE reference identifier).
- How effective it is at detecting previously unknown malicious events, such as new attacks or variants on existing attacks, without reconfiguring or updating the IDPS. Anomaly detection and stateful protocol analysis techniques typically perform better than signature-based detection techniques in recognizing unknown events.
- How effective it is at detecting known and unknown malicious events that have been concealed through evasion techniques. Examples of such techniques include unusual IP packet fragmentation, non-standard application port use, and alternate character sets or other character encoding.

[54] More information on CVE is available from NIST SP 800-51, *Use of the Common Vulnerabilities and Exposures (CVE) Vulnerability Naming Scheme*, which is located at http://csrc.nist.gov/publications/nistpubs/, and the CVE Web site at http://cve.mitre.org/.

- How accurately it can determine the success or failure of attacks.
- What response mechanisms it offers, excluding prevention responses (which are covered in Section 9.2.4). Examples include logging events (both locally and to remote log servers), displaying console alerts, and sending Simple Network Management Protocol (SNMP) traps, e-mails, text messages, and pages. The criterion also includes effective prioritization of events, such as taking different actions when a certain type of event occurs or when an event involves a certain system or service.
- How administrators can customize detection capabilities by modifying signatures, policies, and other settings. Examples include altering whitelists, blacklists, and thresholds; customizing code to reduce false positives or false negatives; and writing new signatures or policies from scratch or based on samples or frameworks. Evaluators should consider how easily the customizations can be performed (e.g., through a GUI, through editing text files). If the customizations require knowledge of a programming language, additional considerations include the following:
  - Is the language commonly used or is it a specialty/proprietary language that administrators would need to learn?
  - How complex and powerful is the language?
  - Does the product offer a development environment or other tools to assist in customization, such as syntax checking or virtual machines for testing customizations before implementing them?
  - When the product is updated or upgraded, how are code customizations preserved?
- How effectively the product can use data from other sources, such as vulnerability scan results and logs from other IDPSs, to correlate events and improve the prioritization of alerts.

9.2.4 Prevention Capabilities

Organizations should determine whether or not the IDPS solution may need to perform prevention actions, including future needs, and evaluate the prevention capabilities of each candidate product. Most prevention capabilities are specific to a particular type of IDPS; information on common capabilities is presented in Sections 4 through 7 for each IDPS product type. When available, it is generally preferred to have a product that has multiple prevention capabilities instead of only one, because some methods are more effective than others in certain situations and ineffective in others. All IDPS products should offer considerable granularity in configuration options for prevention methods, such as enabling or disabling them only for particular alerts, suppressing prevention methods for hosts on whitelists, and allowing administrators to specify which prevention method should be used for each alert if multiple methods are available. Some products offer additional granularity that may be beneficial, such as performing prevention actions only if a certain system is being attacked.

9.3 Performance Requirements

Comparing the performance of IDPS products is challenging for the following reasons:

- Performance is highly dependent on the configuration and tuning of each product. Although testing can be performed using the default settings of each product, some products are designed with the assumption that they will need extensive customization and tuning.
- Performance and detection are often at odds; having more complex and robust detection capabilities often causes poorer performance because they require more processing capability and memory.

- Many IDPS components are appliance-based and have many hardware models and configurations available, each with its own performance characteristics. Other IDPS components are not appliance-based, so their hardware, OSs, and OS configurations may vary widely, which can all affect performance.
- There are no open standards for performance testing, nor are there publicly available, comprehensive, up-to-date test suites.

Accordingly, evaluators should focus on the general performance characteristics of IDPS products and avoid differentiating products by slight differences in reported performance capabilities. Vendors typically rate their products by maximum capacity, such as the volume of network traffic or number of packets per second monitored for network-based IDPS, the number of events monitored per second for host-based IDPS, or the flows monitored per second or the number of hosts that can be profiled for NBA systems. Section 9.6 provides guidance on collecting data on IDPS performance as part of an evaluation. When evaluating maximum capacity claims, evaluators should consider the following questions:

- Does the maximum capacity reflect activity that is being analyzed or activity that is being monitored but not necessarily analyzed? For example, a network-based IDPS might perform little or no analysis on the use of certain application protocols.
- What was the nature of the activity used to measure capacity? This knowledge can help evaluators to determine if the testing used an environment similar to their own or had significant differences that could affect performance results. Aspects of this to consider include the following:
  - How was the activity used for testing generated?
  - What types of malicious activity were included in the testing? What percentage of the events monitored by the IDPS was malicious? What percentage of the malicious events was detected by the IDPS under maximum load?
  - For network traffic, what protocols were used and in roughly what percentages? For host-based activity, what applications were run, and what other sources of events were used?
  - How closely did the activity used for testing reflect the actual conditions of the production environment?
- How was the IDPS configured? Was the default configuration used? If not, what detection capabilities, logging capabilities, and other features were enabled or disabled from the default?
- For any non-appliance components, what hardware, OSs, and applications or services were in use?
- Who performed the testing?
- When was the testing performed?

Evaluators should also consider the performance features that each IDPS under consideration offers. Possible considerations for performance features include the following:

- Does the IDPS offer any performance tuning features, either manually configured or automatically implemented? For example, if an IDPS is being overwhelmed by high volumes of activity, can it alter its detection capabilities so that it temporarily performs less extensive analysis on all the traffic or stops analyzing low-risk traffic?

- For products that track state (e.g., stateful protocol analysis of network connections), how many activities (e.g., connections) can they track state for simultaneously? How long is state information maintained normally and under maximum load?
- For products that process the actual events, not copies of the events (e.g., inline network-based IDPS sensors), how much latency does the processing cause? For example, there might be a delay of 50 microseconds between when a network-based IDPS sensor receives a packet and when the IDPS retransmits that packet to continue to its destination. A host-based IDPS might delay the execution of system calls for a similarly short time. Under high loads, IDPS products might experience significantly higher latency, so it is important to consider latency under both typical and extreme loads.
- For products that process copies of events, not the actual events (e.g., passive network-based IDPS sensors, NBA software analyzing network flow logs sent by routers), how long does it take from the occurrence of an event to the event's detection and reporting by the IDPS?

9.4 Management Requirements

Evaluating the management capabilities of each IDPS product is very important because if a product is hard to manage or does not offer the necessary management functionality, then it is likely that the product will not be used as effectively as originally intended. This section presents IDPS management capability considerations in three categories:

- Design and implementation
- Operation and maintenance
- Training, documentation, and technical support.

Section 9.6 provides guidance on collecting data on IDPS management capabilities as part of an evaluation.

9.4.1 Design and Implementation

Most aspects of IDPS design and implementation are specific to each IDPS technology type; Sections 4 through 7 contain detailed information on design and implementation considerations. In addition to those, organizations should also consider general criteria related to reliability, interoperability, scalability, and security.

9.4.1.1 Reliability

Organizations should ensure that the IDPS products they select will be sufficiently reliable to meet their requirements. Possible considerations for reliability include the following:

- What types of redundant hardware are included or available separately for appliances, such as duplicate power supplies, network interface cards, storage devices (e.g., hard drives, flash ROMs), and CPUs?
- What software redundancy features are incorporated into the products, especially for agents and sensors, such as the product automatically restarting itself and/or supporting services when they fail?
- Can the product use multiple management servers so that if one fails, sensors or agents automatically fail over to another one? How disruptive is the failover process?

- Can multiple sensors be deployed to monitor the same activity so that if one fails, another automatically assumes its responsibilities? How disruptive is the failover process (e.g., loss of state tracking, loss of event counts for thresholds)?
- If a sensor fails, how easily can its configuration be transferred to another sensor (e.g., transferring a sensor CD and configuration floppy from the first sensor to the second sensor, then rebooting the second sensor)?

9.4.1.2 Interoperability

Organizations should ensure that the IDPS products they select will interoperate effectively with the desired systems. These systems could include the following:

- Data input sources, such as other IDPS products, log files, and vulnerability scanning results
- Log analysis and management software, such as syslog and other logging servers, SIEM software, and network management software
- Systems to be reconfigured by prevention actions, such as firewalls and routers.

9.4.1.3 Scalability

When evaluating IDPS products, organizations should consider not only their current needs, but also possible future needs, so that they choose products that are sufficiently scalable. Possible considerations for scalability include the following:

- The number of sensors or agents, management servers, consoles, and other IDPS components that can be part of a single logical implementation
- The number of sensors or agents that a single management server can support
- The range of appliances available for appliance-based IDPS components (e.g., appliance devices with varying capacities), and the ability to expand appliances (e.g., add more memory, network interface cards [NIC], or storage devices)
- How multiple sensors or agents can share monitoring functions for a network or system, including how load balancing can be performed with or without the use of separate load balancing devices
- How many networks a network-based, wireless, or NBA sensor can monitor simultaneously; how many network interfaces a host-based agent can monitor simultaneously
- How the IDPS's storage capabilities can be expanded and enhanced (e.g., automated archival of older data, use of separate storage devices)
- What levels of activity (e.g., network traffic, system calls, log entries) each of the IDPS components can support
- How well the IDPS solution integrates the management and monitoring of multiple sensors or agents, management servers, and other components
- The cost of and resources needed for each scalability option.

9.4.1.4 Security

When evaluating IDPS products, organizations should consider the security requirements for the IDPS solution itself. Evaluators should review the security controls listed in NIST SP 800-53, *Recommended Security Controls for Federal Information Systems* and consider which controls should be included in the IDPS security criteria. Examples of security considerations include the following:

- How stored data (including logs) and communications among all the IDPS components are protected, such as using alternate data channels or FIPS-approved encryption and digital signature algorithms to support data confidentiality and integrity when needed
- The authentication, access control, and auditing features performed for IDPS usage and administration
- The IDPS's resistance to attacks against it, such as blinding and DoS attacks.

9.4.2 Operation and Maintenance

This criterion focuses on requirements for the user and administrator interfaces for ongoing management of the IDPS. This includes the ease of performing daily monitoring, analysis, and reporting activities; managing and maintaining the IDPS; and applying updates. Possible specific criteria for each of these areas is provided below. In addition, evaluators should consult with vendors, analysts, and/or trusted peers to determine the level of technical and security expertise needed to use and maintain each product. Evaluators should ask vendors what their assumptions are regarding the users and administrators of their products.

9.4.2.1 Daily Use

Organizations should consider how the IDPS solution needs to be used on a daily basis for monitoring security events, performing analysis of events of interest, and generating reports. Because these three activities are often intertwined, it is often easiest to assess them together. Daily use considerations for IDPSs should include the following:

- How it displays events and alerts to users, what features it provides to ease analysis (e.g., drill-down capability, links to supporting information, correlation of events from multiple sensors or agents, color-coding alerts to indicate their severity/priority), and how users can customize the views and filters to alter the display of events and alerts
- How it displays its status information to users and administrators (e.g., how a sensor failure is communicated)
- How it notifies users and administrators of both serious security events and IDPS failures and other operational problems
- How much supporting information it records for events (e.g., is enough information recorded to allow analysts to determine what happened?)
- How many interfaces/programs are needed for the daily use functions (e.g., can a single GUI provide all the functions that the IDPS users need?)
- How many concurrent interfaces are supported

- What default report formats are offered (e.g., text, comma-separated values [CSV], HTML, Extensible Markup Language [XML], PDF, Microsoft Word, Microsoft Excel) and what data storage formats are supported for IDPS data, log, and report retention
- How reports can be customized (both altering existing reports and creating new reports)
- Whether or not reports can be generated automatically (e.g., on a schedule, when certain events occur), how the reports can be distributed (e.g., e-mailed to administrators), and how the distributed reports are protected (e.g., file encryption)
- Whether or not it offers any workflow tracking capabilities, such as incident tracking.

9.4.2.2 Maintenance

Organizations should consider how the IDPS solution and its components should be maintained, and then evaluate products based on those maintenance requirements. Maintenance considerations should include the following:

- Whether or not sensors or agents can be managed both independently and through a management server, and whether such accesses are logged
- What local and remote maintenance mechanisms are available (e.g., locally installed GUI, Web-based console, command-line interface [CLI], third-party tools), and what differences there are (if any) in their functionality
- Which components can be maintained locally and remotely with each maintenance mechanism
- What security protections are provided for each maintenance mechanism (e.g., strong encryption for network traffic)
- How component configuration settings can be backed up and restored, and how they can be transferred from a component to a replacement component (e.g., swapping sensor appliances because of hardware failure)
- How robust the product is at logging component status information (e.g., low disk space, high CPU utilization), operational failures, and other events that may necessitate maintenance actions
- Whether or not the IDPS provides sufficiently robust log management tools, and if not, how administrators could compensate (e.g., write scripts, acquire third-party tools).

9.4.2.3 Updates

Organizations should carefully consider how the vendor of each evaluated IDPS product releases updates for it. Aspects of this to consider include the following:

- How often regular major and minor updates to each component are released (e.g., sensors, management servers, consoles)
- How often updates to detection capabilities are released in response to major new threats, and how soon after the identification of a new threat the corresponding update is typically available
- Which types of updates usually or sometimes require that IDPS components be rebooted or restarted

- How the organization receives each type of update from the vendor (e.g., sensor upgrade distributed on CD, signature updates available for download through the console or from the vendor's technical support Web site)
- How the authenticity and integrity of updates can be confirmed (e.g., through cryptographic checksums)
- How updates can be distributed to IDPS components such as sensors and consoles (e.g., automated process, manual installation)
- How the installation of updates can affect existing IDPS settings or customizations.

9.4.3 Training, Documentation, and Technical Support

Organizations should consider the resources available to the IDPS administrators and users for learning about the IDPS's functionality and characteristics and for receiving assistance when problems occur. These resources—training, documentation, and technical support—should take into account both administrator and user needs, as well as different experience levels.

- **Training.** Most IDPS vendors offer training classes for their products. Some offer a single class per product, while others offer separate classes for users and administrators. Separate classes may also be available for particular IDPS components, such as consoles or management servers, or for specialized tasks such as code customization or report creation. Some vendors also offer general IDPS classes that are intended to give users a better understanding of IDPS principles. Third parties also offer general IDPS classes and classes for some specific IDPS products. Organizations should consider which training classes are available that meet their needs, what format the classes are in (e.g., instructor-led, online, computer-based training [CBT]), and where the classes are held (e.g., the IDPS vendor's headquarters, regional locations, the customer's site). For instructor-led classes, organizations should determine if they include lab work or other hands-on exercises that allow users to use the actual IDPS equipment.
- **Documentation.** IDPS products usually include documentation in paper or electronic forms. Examples include installation, user, administrator, and signature/policy development guides. Electronic guides are often fully searchable; some products also offer context-sensitive help through the console, allowing a user to easily access the pertinent documentation for a particular console feature or security event type. If guides are provided on paper only, organizations should determine if the guides can be duplicated, and if not, what the availability of additional copies is.
- **Technical Support.** Most IDPS vendors offer multiple technical support contracts. For example, one contract might provide basic phone, e-mail, and Web-based support during business hours with a one-hour response time, while another contract might provide 24-hour access to senior support staff with a 15-minute response time and include annual onsite visits and consulting services. Organizations should take care to determine what activities are and are not covered by a contract; for example, tuning and customization, such as writing signatures or customizing reports, might not be included. Vendors typically provide multiple support contract options so that each customer can select one that is cost-effective for them. Free technical support is also available for some products through user groups, mailing lists, forums, and other methods.

9.5 Life Cycle Costs

Organizations should compare the funding they have available for IDPS solutions to the estimated life cycle costs for each of the evaluated solutions. Quantifying the life cycle costs for IDPS solutions can be difficult because there are many environment-specific factors that impact cost, and because it is usually

challenging to capture the cost benefits provided by IDPSs. The criteria presented below focus on the basic costs of the IDPS solution itself and do not take into account any cost savings achieved by IDPS use.

- **Initial Costs.** The initial costs of acquiring and deploying a solution typically include the following:
  - Hardware, including appliances, additional network equipment (e.g., management network, network taps, IDS load balancers), and hosts for non-appliance components (e.g., consoles)
  - Software and software licensing fees for IDPS components and supporting software (e.g., reporting tools, database software)
  - Installation and initial configuration costs, which could include external assistance as well as internal labor
  - Customization costs, such as having programmers develop custom scripts or reports
  - Training costs, if the necessary training is not included as part of the initial hardware and software purchase.
- **Maintenance Costs.** Expected maintenance costs for IDPS solutions typically include the following:
  - Labor. This includes the cost of staff performing IDPS administration and analysis.
  - Software licensing fees, subscription fees, or maintenance contracts. These costs, typically incurred on an annual basis, usually provide the purchaser with IDPS software and signature updates.
  - Technical support fees. Many organizations purchase technical support contracts for their IDPS products; these contracts are typically annual. Some organizations pay a fee per technical support call instead of an annual contract.
  - Training costs. Training might be needed periodically in preparation for deploying new versions of an IDPS product, as well as for new IDPS users and administrators. Organizations might want to have customized training classes that focus on the elements of the IDPS product that are most important to the organization, and also take into account certain aspects of the organization's environment and needs.
  - Customization costs. During the use of an IDPS product, users and administrators might need the product to be further customized, such as having programmers develop additional custom reports or modify existing reports, and having programmers or administrators create custom analyzers and signatures.
  - Professional services or technical support that falls outside the technical support contract. Examples include designing IDPS implementations, performing product installations, tuning sensors or agents, creating and customizing reports, and assisting with incident response efforts. Organizations can perform these services themselves, or they can purchase services from IDPS vendors and third parties.

9.6 Evaluating Products

After collecting requirements and selecting criteria, evaluators need to find sources of information about the products to be evaluated. Common product data sources include the following:

- Test lab or real-world environment testing of selected IDPS products
- Previous real-world experience with IDPSs from individuals within the organization and trusted individuals at other organizations
- Vendor-provided information, such as product manuals and datasheets, whitepapers, product demonstrations, and discussions with vendor employees
- Third-party product reviews, including reviews of individual products and comparisons of multiple products.

Section 9.6.1describes the challenges in performing IDPS product testing as part of an evaluation. Section 9.6.2 presents recommendations for using the data sources described above when conducting an evaluation.

9.6.1 IDPS Testing Challenges

An organization performing its own in-depth hands-on testing of IDPS products ideally could generate comprehensive data on the products that would accurately reflect how well-suited each product is to meeting the organization's needs. However, this is generally not feasible to achieve because of how difficult and resource-intensive it is to perform IDPS testing well. The following are some of the major reasons for these problems:[55]

- **Test Methodology.** There is no standard methodology for performing IDPS testing. Also, details are not available for most of the methodologies used for commercial evaluation of IDPS products. Organizations performing IDPS testing need to create their own methodologies or perform a survey of existing methodologies, determine which would be best for their needs, and then design and implement testing processes using the selected methodology. Also, a different methodology, including test environments and test suites, is needed for each type of IDPS technology.
- **Multiple Environments.** Organizations performing IDPS testing should conduct it in both real-world and lab environments. The real-world testing helps evaluators to understand how well the product will likely function in their environment. The lab testing allows evaluators to better assess the detection and prevention capabilities of the product. Detection results can be difficult to understand when real-world activity is being monitored because the real-world activity is likely to contain different types of malicious activity, and it is sometimes unclear whether or not the detected activity was actually malicious. Prevention capabilities are generally not tested in real-world environments because they can easily cause disruptions to benign activity. It is very difficult to duplicate real-world environments in lab environments, so organizations performing IDPS testing generally need to do their testing separately in each environment.
- **Test Availability.** There are no standard IDPS test suites available. Organizations performing IDPS testing need to find ways to generate both malicious activity (to see how well the products identify them) and benign activity (to put the product under normal or heavy loads). The malicious activity should accurately reflect the composition of recent threats against the organization's systems and networks; accordingly, it can take considerable time to identify those threats and acquire tests for them. The tests also need to take into account all detection methodologies used by the IDPSs, because usually different types of tests are needed to properly evaluate the effectiveness of each

[55] For more information on the challenges of IDPS testing, see NIST Interagency Report (IR) 7007, *An Overview of Issues in Testing Intrusion Detection Systems.* It is available at http://csrc.nist.gov/publications/nistir/nistir-7007.pdf. Although it is focused primarily on testing network-based IDPSs, most of the testing problems it discusses are applicable to testing any type of IDPS technology.

methodology.[56] Typically it takes a combination of carefully selected tools and custom-written attack scripts to build a reasonable test suite. Each tool and script should be reviewed and tested to ensure that it performs the tests properly.

- **Lab Environment Resources.** Organizations performing IDPS testing in lab environments typically need to expend considerable resources in setting up the lab environments. Attacker and victim systems need to be set up and configured. The victim systems need to run the OSs, services, and applications targeted by the attacks. Depending on the methodologies used by the IDPSs, the victim systems may need to have all the vulnerabilities exploited by the attacks. Some IDPSs might alert only on attacks that they think will be successful; also, some attacks will stop executing if they do not detect exploitable vulnerabilities. Evaluators also need to be aware of the capabilities of the IDPSs; for example, an IDPS might see a few attacks from a single attacker system and automatically perform prevention actions to stop all future attacks from that system.

- **Product Equivalence.** Most IDPS products need to be tuned and customized to meet the requirements of the organization. Each product is configured somewhat differently by default, so organizations performing IDPS testing should attempt to tune and customize the products so that they are as similar as possible. For example, thresholds such as the number of failed login attempts permitted in a certain time period should be set to the same values. Also, each detection feature should be enabled or disabled consistently on all the IDPSs. This is often very difficult to accomplish—for example, a product performing signature-based detection tends to have settings based on specific exploits being performed, while a product performing stateful protocol analysis detection often has settings based on specific vulnerabilities being exploited. Evaluators would need to map the exploits and vulnerabilities to determine the equivalent settings on different IDPSs.

9.6.2 Recommendations for Performing IDPS Evaluations

The challenges in performing in-depth hands-on IDPS testing often make it infeasible; however, performing some amount of IDPS testing is generally quite helpful in evaluating how well IDPSs meet an organization's requirements for security capabilities, performance, and operation and maintenance. IDPS testing is also helpful in setting realistic expectations for the capabilities of the products and the amount of labor required to maintain and monitor them in the organization's environment. Accordingly, organizations should consider using a combination of several data sources, such as limited product testing, vendor-provided information, third-party product reviews, and individuals' previous IDPS experience, when performing IDPS product evaluations. For example, organizations could use data sources other than product testing to narrow the product selection to only a few choices, and then perform limited testing of those choices only. In some cases, omitting product testing and performing a paper-only evaluation of a product is necessary because of time and resource constraints, but generally an evaluation will produce better results if it incorporates at least some product testing.

When using data from other parties, organizations should consider the fidelity of the data. Data is often presented without a detailed explanation of how it was created, such as maximum capacities or detection accuracy rates. Because there are no standard methodologies for compiling such data, organizations should be cautious when comparing data from different sources, because the measurements may have been performed using fundamentally different methods.

When performing hands-on IDPS testing, organizations should focus on those testing methods that are most likely to be valuable. Testers should also avoid disrupting the organization's operations. The following provides guidance on performing testing for each class of IDPS product. After testing has been

[56] Another complicating factor is that it is often not apparent which methodologies particular products use, so it might be difficult to determine which types of tests are needed.

completed, testers should ensure that any hardware on loan from IDPS vendors has its writable media sanitized appropriately to remove the organization's data.[57]

9.6.2.1 Network-Based

Valuable insights into network-based IDPS security capabilities (especially detection accuracy and tuning), performance with the organization's network traffic, and the operation and maintenance of the IDPS can be gained by performing real-world testing of the IDPS. However, it is generally prudent to keep the IDPS somewhat separate from the production environment during this testing so that the IDPS does not adversely affect it (e.g., increase latency) and so that any vulnerabilities in the IDPS cannot be exploited by attackers. An IDS load balancer is ideal for giving multiple sensors identical copies of the network traffic simultaneously, allowing for side-by-side comparisons of the products, while isolating the sensors and preventing them from inadvertently disrupting production (traffic passes through a load balancer in only one direction). Depending on the network architecture, it may be possible to test sensors in inline deployments by duplicating traffic at the network locations where each of an inline sensor's network interfaces would be and feeding that traffic to the inline sensors' interfaces. Otherwise, most inline sensors can be placed into a passive mode and tested as passive; the benefit of testing them with production traffic in inline mode is to study their performance.

Lab testing of network-based IDPSs is most beneficial for evaluating the following:

- **The prevention capabilities of products.** Testers can set up test systems (targets and attacking systems), generate attacks, and monitor the effectiveness of each IDPS's prevention actions.
- **The performance of inline sensor deployments.** If this cannot be done as part of real-world testing, testers could use network traffic generation tools or replay previously recorded traffic to generate activity to pass through the sensor.
- **Design and implementation-related characteristics.** Product reliability could be tested by deploying multiple sensors or management servers, configuring them for failover conditions, generating traffic for them to process, and then intentionally causing a failure of one component and monitoring the resulting product behavior. Interoperability could be tested by configuring test systems representing the products with which the IDPS must interoperate, and then generating activity that should cause the products to work together. The security of the IDPS itself can also be tested through vulnerability scanning, penetration testing, and other methods.

9.6.2.2 Wireless

The methods to be used for testing wireless IDPSs should be selected primarily by the format of the wireless IDPS sensors to be tested:

- **Mobile sensors, fixed sensors, and sensors bundled with APs.**[58] Testing of security capabilities, performance, and some facets of operation and maintenance can typically be performed by using the sensors in production environments, with the caveat that prevention capabilities should be disabled. Prevention capabilities could be evaluated in an isolated test environment that is out of range of all other wireless local area networks. This test environment would contain test access points and test wireless clients using the access points; testers might need to set up test systems that the wireless

[57] For more information on media sanitization, see NIST SP 800-88, *Guidelines for Media Sanitization*, which is available at http://csrc.nist.gov/publications/nistpubs/.

[58] For sensors bundled with APs, these test instructions assume that a non-production AP is used for the testing. Deploying sensor software onto production APs for testing purposes is not recommended because it could disrupt the production environment.

clients can access to generate wireless network communications. Attacks can be issued from one or more wireless clients, and rogue access points can be deployed in the test environment. If the sensors will be integrated with an IDPS infrastructure, any testing of this should also be performed in the test environment to evaluate performance, operation and maintenance, and design and implementation characteristics without jeopardizing the production infrastructure (e.g., an IDPS sensor could have vulnerabilities that could be exploited by attackers within range of the sensor).

- **Sensors bundled with wireless switches.** Generally, this testing should be performed by setting up a test switch with sensor software in a test environment like the one described above for other types of wireless sensors. The same type of testing described above should be performed.

9.6.2.3 NBA

If the NBA products will be directly monitoring network traffic, then real-world and lab testing of that capability should be performed based on the guidance given for testing network-based IDPSs. If the NBA products will be monitoring network flow logs from other devices, the preferred method for real-world testing of that capability is to set up a separate network and forward the logs from the devices over that network to the NBA sensors. This protects the NBA solution and allows the bandwidth used by the solution to be measured easily. If the production networks will be used instead of a separate network, testers need to be very careful not to overwhelm the production networks with the volume of logs, particularly if multiple NBA products are being tested simultaneously. Testing can also be performed in a lab environment by providing copies of production logs to the NBA products. NBA product lab testing is also beneficial for the same reasons cited for network-based IDPS lab testing: evaluating prevention capabilities, inline sensor performance, and product design and implementation-related characteristics.

9.6.2.4 Host-Based

Host-based IDPSs are typically more challenging to perform real-world testing for than any other type of IDPS. Agents alter the hosts that they monitor and can adversely affect their performance and functionality (e.g., IDPS shims interfering with other applications); appliance-based IDPSs are deployed inline in front of production systems. The methods to be used for testing host-based IDPSs should be selected primarily by the roles of the hosts to be protected:

- **A server (including a single application service on a server).** Testing should be performed in a test environment only. For example, a test server could be created that mimics a production server or even uses one of its backups. Typical activity directed at the server, both benign and malicious, should be generated by test systems (e.g., scripts or tools to create HTTP requests) and monitored by the host-based IDPS. Testers can perform attacks against the server and monitor the prevention actions performed without endangering any production systems. Testers can also measure the impact of the host-based IDPS on the performance of the server and evaluate the reliability and security of the host-based IDPS by attempting to disrupt it.

- **A client host (desktop or laptop).** Initial testing should be performed in a test environment to identify major performance and functionality problems that host-based IDPSs might introduce. The reliability and security of the IDPS can also be evaluated in a test environment. Testing of agents' security capabilities, prevention actions, and other characteristics can be conducted in both a test environment and a production environment because the risk posed by IDPS failure to the production environment is very low. Attacks should only be issued against the hosts in a test environment, while the agents' behavior against benign activity can be tested most easily in a real-world environment. For example, a few of the testers might volunteer to have IDPS agents installed on their production desktops and document the agents' behavior and any problems they cause for a week or two. This provides true real-world testing of the agents. For agents that necessitate user interaction, such as

responding to queries about permitting or denying activity, conducting end user testing in a test or production environment is also prudent.

When testing host-based IDPSs, organizations should test the most commonly used and important OSs and applications that need to be protected. The architecture of each OS and each application is different, so a single product might exhibit significantly different behavior when used on different platforms.

9.7 Summary

Before evaluating IDPS products, organizations should first define the general requirements that the products should meet. The features provided by IDPS products and the methodologies that they use vary considerably, so a product that best meets one organization's requirements might not be suitable for meeting another organization's requirements. Evaluators first need to understand the characteristics of the organization's system and network environments and plans for near-term changes, so that an IDPS can be selected that will be compatible with them and able to monitor the events of interest on the systems and/or networks. This knowledge is also needed to design the IDPS solution. After gaining an understanding of the existing system and network environments, evaluators should articulate the goals and objectives they wish to attain by using an IDPS. Evaluators should also review their existing security and other IT policies before selecting products. The policies serve as a specification for many of the features that the IDPS products need to provide. In addition, evaluators should understand whether or not the organization is subject to oversight or review by another organization. If so, they should determine if that oversight authority requires IDPSs or other specific system security resources. Resource constraints should also be taken into consideration by evaluators.

In addition to defining general requirements, evaluators also need to define more specialized sets of requirements:

- Security capabilities, including information gathering, logging, detection, and prevention
- Performance, including maximum capacity and performance features
- Management, including design and implementation, operation and maintenance, and training, documentation, and technical support
- Life cycle costs, both initial and maintenance costs.

Organizations could use these criteria as a basis for creating an organization-specific set of criteria that takes into account an organization's environment, policies, and existing security and network infrastructure. After collecting requirements and selecting criteria, evaluators need to find viable sources of information about the products to be evaluated. Common product data sources include test lab or real-world product testing, vendor-provided information, third-party product reviews, and previous IDPS experience from individuals within the organization and trusted individuals at other organizations.

There are major challenges in performing in-depth hands-on IDPS testing with satisfactory results, which often make it infeasible. Most organizations find the results of limited IDPS testing helpful for evaluating daily use, interoperability, and security requirements. Organizations should consider using a combination of several data sources when performing IDPS product evaluations. When using data from other parties, organizations should consider the fidelity of the data because it is often presented without an explanation of how it was generated. When performing hands-on IDPS testing, organizations should focus on those testing methods that are most likely to be valuable and should avoid methods that are more likely to disrupt the organization's operations.

Appendix A—Glossary

Selected terms used in the *Guide to Intrusion Detection and Prevention Systems (IDPS)* are defined below.

Agent: A host-based intrusion detection and prevention program that monitors and analyzes activity and may also perform prevention actions.

Alert: A notification of an important observed event.

Anomaly-Based Detection: The process of comparing definitions of what activity is considered normal against observed events to identify significant deviations.

Antivirus Software: A program that monitors a computer or network to identify all major types of malware and prevent or contain malware incidents.

Application-Based Intrusion Detection and Prevention System: A host-based intrusion detection and prevention system that performs monitoring for a specific application service only, such as a Web server program or a database server program.

Blacklist: A list of discrete entities, such as hosts or applications, that have been previously determined to be associated with malicious activity.

Blinding: Generating network traffic that is likely to trigger many alerts in a short period of time, to conceal alerts triggered by a "real" attack performed simultaneously.

Channel Scanning: Changing the channel being monitored by a wireless intrusion detection and prevention system.

Console: A program that provides user and administrator interfaces to an intrusion detection and prevention system.

Database Server: A repository for event information recorded by sensors, agents, or management servers.

Evasion: Modifying the format or timing of malicious activity so that its appearance changes but its effect on the target is the same.

False Negative: An instance in which an intrusion detection and prevention technology fails to identify malicious activity as being such.

False Positive: An instance in which an intrusion detection and prevention technology incorrectly identifies benign activity as being malicious.

Flooding: Sending large numbers of messages to a host or network at a high rate. In this publication, it specifically refers to wireless access points.

Flow: A particular network communication session occurring between hosts.

Host-Based Intrusion Detection and Prevention System: A program that monitors the characteristics of a single host and the events occurring within that host to identify and stop suspicious activity.

Incident: A violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.

Inline Sensor: A sensor deployed so that the network traffic it is monitoring must pass through it.

Intrusion Detection: The process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents.

Intrusion Detection and Prevention: The process of monitoring the events occurring in a computer system or network, analyzing them for signs of possible incidents, and attempting to stop detected possible incidents. See also "intrusion prevention".

Intrusion Detection System Load Balancer: A device that aggregates and directs network traffic to monitoring systems, such as intrusion detection and prevention sensors.

Intrusion Detection System: Software that automates the intrusion detection process.

Intrusion Prevention: The process of monitoring the events occurring in a computer system or network, analyzing them for signs of possible incidents, and attempting to stop detected possible incidents. See also "intrusion detection and prevention".

Intrusion Prevention System: Software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. Also called an intrusion detection and prevention system.

Jamming: Emitting electromagnetic energy on a wireless network's frequencies to make them unusable by the network.

Malware: A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or of otherwise annoying or disrupting the victim.

Management Network: A separate network strictly designed for security software management.

Management Server: A centralized device that receives information from sensors or agents and manages them.

Network-Based Intrusion Detection and Prevention System: An intrusion detection and prevention system that monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify and stop suspicious activity.

Network Behavior Analysis System: An intrusion detection and prevention system that examines network traffic to identify and stop threats that generate unusual traffic flows.

Network Tap: A direct connection between a sensor and the physical network media itself, such as a fiber optic cable.

Passive Fingerprinting: Analyzing packet headers for certain unusual characteristics or combinations of characteristics that are exhibited by particular operating systems or applications.

Passive Sensor: A sensor that is deployed so that it monitors a copy of the actual network traffic.

Promiscuous Mode: A configuration setting for a network interface card that causes it to accept all incoming packets that it sees, regardless of their intended destinations.

Sensor: An intrusion detection and prevention system component that monitors and analyzes network activity and may also perform prevention actions.

Shim: A layer of host-based intrusion detection and prevention code placed between existing layers of code on a host that intercepts data and analyzes it.

Signature: A pattern that corresponds to a known threat.

Signature-Based Detection: The process of comparing signatures against observed events to identify possible incidents.

Spanning Port: A switch port that can see all network traffic going through the switch.

Stateful Protocol Analysis: The process of comparing predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations.

Stealth Mode: Operating an intrusion detection and prevention sensor without IP addresses assigned to its monitoring network interfaces.

Threshold: A value that sets the limit between normal and abnormal behavior.

Triangulation: Identifying the physical location of a detected threat against a wireless network by estimating the threat's approximate distance from multiple wireless sensors by the strength of the threat's signal received by each sensor, then calculating the physical location at which the threat would be the estimated distance from each sensor.

Tuning: Altering the configuration of an intrusion detection and prevention system to improve its detection accuracy.

Whitelist: A list of discrete entities, such as hosts or applications, that are known to be benign.

Wireless Intrusion Detection and Prevention System: An intrusion detection and prevention system that monitors wireless network traffic and analyzes its wireless networking protocols to identify and stop suspicious activity involving the protocols themselves.

This page has been left blank intentionally.

Appendix B—Acronyms

Selected acronyms used in the *Guide to Intrusion Detection and Prevention Systems (IDPS)* are defined below.

AP Access Point
ARP Address Resolution Protocol

CAIDA Cooperative Association for Internet Data Analysis
CIAC Computer Incident Advisory Capability
CLI Command-Line Interface
CMVP Cryptographic Module Validation Program
COM Component Object Model
CPU Central Processing Unit
CSIRT Computer Security Incident Response Team
CSRC Computer Security Resource Center
CSV Comma Separated Values
CVE Common Vulnerabilities and Exposures

DDoS Distributed Denial of Service
DHCP Dynamic Host Configuration Protocol
DLL Dynamic Link Library
DMZ Demilitarized Zone
DNS Domain Name System
DoS Denial of Service
DS Distribution System
DShield Distributed Intrusion Detection System

EICAR European Institute for Computer Antivirus Research
ESP Encapsulating Security Payload

FIPS Federal Information Processing Standards
FISMA Federal Information Security Management Act
FTP File Transfer Protocol

GHz Gigahertz
GUI Graphical User Interface

HTTP Hypertext Transfer Protocol
HTTPS Hypertext Transfer Protocol over SSL

ICMP Internet Control Message Protocol
IDPS Intrusion Detection and Prevention System
IDS Intrusion Detection System
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IGMP Internet Group Management Protocol
IM Instant Messaging
IMAP Internet Message Access Protocol
IP Internet Protocol
IPS Intrusion Prevention System

IPsec	Internet Protocol Security
IRC	Internet Relay Chat
ISC	Internet Storm Center
IT	Information Technology
ITL	Information Technology Laboratory
LAN	Local Area Network
MAC	Media Access Control
NBA	Network Behavior Analysis
NBAD	Network Behavior Anomaly Detection
NFAT	Network Forensic Analysis Tool
NFS	Network File System
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NVD	National Vulnerability Database
OMB	Office of Management and Budget
OS	Operating System
PDA	Personal Digital Assistant
PoE	Power over Ethernet
POP	Post Office Protocol
RF	Radio Frequency
RFC	Request for Comment
ROM	Read-Only Memory
RPC	Remote Procedure Call
SEM	Security Event Management
SIEM	Security Information and Event Management
SIM	Security Information Management
SIP	Session Initiation Protocol
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SP	Special Publication
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
STA	Station
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TTL	Time to Live
UDP	User Datagram Protocol

USB	Universal Serial Bus
US-CERT	United States Computer Emergency Readiness Team
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WVE	Wireless Vulnerabilities and Exploits
XML	Extensible Markup Language

This page has been left blank intentionally.

Appendix C—Tools and Resources

The lists below provide examples of tools and resources that may be helpful.

Print Resources

Bace, Rebecca, *Intrusion Detection,* Macmillan Technical Publishing, 2000.

Bejtlich, Richard, *Extrusion Detection,* Addison-Wesley, 2005.

Bejtlich, Richard, *The Tao of Network Security Monitoring: Beyond Intrusion Detection,* Addison-Wesley, 2004.

Crothers, Tim, *Implementing Intrusion Detection Systems: A Hands-On Guide for Securing the Network,* 2002.

Endorf, Carl et al, *Intrusion Detection and Prevention,* McGraw-Hill Osborne Media, 2003.

Kruegel, Chris et al, *Intrusion Detection and Correlation: Challenges and Solutions,* Springer, 2004.

Nazario, Jose, *Defense and Detection Strategies Against Internet Worms,* Artech House Publishers, 2003.

Northcutt, Stephen and Novak, Judy, *Network Intrusion Detection: An Analyst's Handbook, Third Edition,* New Riders, 2003.

Rash, Michael et al, *Intrusion Prevention and Active Response: Deployment Network and Host IPS,* Syngress, 2005.

Organizations

Organization	URL
Computer Incident Advisory Capability (CIAC)	http://www.ciac.org/ciac/
Cooperative Association for Internet Data Analysis (CAIDA)	http://www.caida.org/home/
Distributed Intrusion Detection System (DShield)	http://dshield.org/indexd.html
European Institute for Computer Antivirus Research (EICAR)	http://www.eicar.org/
IETF Intrusion Detection Exchange Format (idwg) Working Group	http://www.ietf.org/html.charters/OLD/idwg-charter.html
Internet Storm Center (ISC)	http://isc.incidents.org/
SANS Institute	http://www.sans.org/
United States Computer Emergency Readiness Team (US-CERT)	http://www.us-cert.gov/
Virus Bulletin	http://www.virusbtn.com/index
Viruslist.com	http://www.viruslist.com/en/
WildList Organization International	http://www.wildlist.org/

Technical Resource Sites

Resource Name	URL
CSRC—Practices & Checklist/Implementation Guides	http://csrc.nist.gov/pcig/cig.html
Unassigned IP Address Ranges	http://www.cymru.com/Documents/bogon-list.html
General and Network-Based IDPS Resources	
An Introduction to Intrusion Detection Systems	http://www.securityfocus.com/infocus/1520
Comparison of Firewall, Intrusion Prevention and Antivirus Technologies	http://www.juniper.net/solutions/literature/white_papers/200063.pdf
Evaluating Intrusion Prevention Systems	http://www.cioupdate.com/article.php/3563306
IDS: Intrusion Detection System	http://www.javvin.com/networksecurity/ids.html
Intrusion Detection System Frequently Asked Questions	http://www.sans.org/resources/idfaq/
Intrusion Detection System Overview	http://www.webopedia.com/TERM/I/intrusion_detection_system.html
Intrusion Detection: Implementation and Operational Issues	http://www.stsc.hill.af.mil/crosstalk/2001/01/mchugh.html
Intrusion Prevention Systems	http://www.nfr.com/resource/downloads/SentivistIPS-WP.pdf
Intrusion Prevention Systems (IPS)	http://www.securecomputing.com/pdf/Intru-Preven-WP1-Aug03-vF.pdf
Intrusion Prevention Systems (IPS)	http://hosteddocs.ittoolbox.com/BW013004.pdf
Intrusion Prevention Systems: the Next Step in the Evolution of IDS	http://www.securityfocus.com/infocus/1670
Recommendations for Deploying an Intrusion-Detection System	http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci781471,00.html
SANS Glossary of Terms Used in Security and Intrusion Detection	http://www.sans.org/resources/glossary.php
State of the Practice of Intrusion Detection Technologies	http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr028.pdf
The Evolution of Intrusion Detection Systems	http://www.securityfocus.com/infocus/1514
Wireless IDPS Resources	
Wireless IDSes Defend Your Airspace	http://www.eweek.com/article2/0,1895,1630842,00.asp
Wireless Intrusion Detection and Response	http://users.ece.gatech.edu/~owen/Research/Conference%20Publications/wireless_IAW2003.pdf
Wireless Intrusion Detection Systems	http://www.securityfocus.com/infocus/1742
Wireless Intrusion Detection Systems: GIAC Security Essentials	http://www.sans.org/rr/whitepapers/wireless/1543.php
NBA IDPS Resources	
Anomaly Detection Can Prevent Network Attacks	http://www.techworld.com/networking/features/index.cfm?featureid=2338&pagtype=samecat
Anomaly Detection in IP Networks	http://users.ece.gatech.edu/~jic/sig03.pdf
Design and Implementation of an Anomaly Detection System: an Empirical Approach	http://luca.ntop.org/ADS.pdf
IDS: Signature Versus Anomaly Detection	http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1092691,00.html?track=IDSLG
Packet vs Flow-Based Anomaly Detection	http://www.esphion.com/pdf/ESP_WP_4_PACKET_V_FLOWS.pdf

Resource Name	URL
The State of Anomaly Detection	http://www.securityfocus.com/infocus/1600
Host-Based IDPS Resources	
Host-Based IDS vs Network-Based IDS	http://www.windowsecurity.com/articles/Hids_vs_Nids_Part1.html
Host-Based IDSs Add to Security Policy	http://www.networkworld.com/news/tech/2003/0915techupdate.html
Host-Based Intrusion Detection System Definition	http://en.wikipedia.org/wiki/Host-based_intrusion_detection_system
Host-Based Intrusion Detection Systems	http://staff.science.uva.nl/~delaat/snb-2004-2005/p19/report.pdf
What Is Host-Based Intrusion Detection?	http://www.sans.org/resources/idfaq/host_based.php

Mailing Lists and Notification Services

Mailing List/Notification Service Name	Location
Incidents	http://www.securityfocus.com/cgi-bin/index.cgi?c=11&op=display_threads&ListID=75&limit=30&offset=0&date=2007-01-16&mode=threaded
Security Focus	http://www.securityfocus.com/ids
SecurityTracker.com	http://securitytracker.com/

Other Technical Resource Documents

Resource Name	URL
IETF, RFC 2267, *Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing*	http://www.ietf.org/rfc/rfc2267.txt
NIST, SP 500-267, *A Profile for IPv6 in the U.S. Government, Version 1.0 (DRAFT)*	http://www.antd.nist.gov/
NIST, SP 800-31, *Intrusion Detection Systems*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-42, *Guideline on Network Security Testing*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-51, *Use of the Common Vulnerabilities and Exposures (CVE) Vulnerability Naming Scheme*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-53, *Recommended Security Controls for Federal Information Systems*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-61, *Computer Security Incident Handling Guide*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-70, *Security Configuration Checklists Program for IT Products*	http://csrc.nist.gov/checklists/
NIST, SP 800-83, *Guide to Malware Incident Prevention and Handling*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-86, *Guide to Integrating Forensic Techniques into Incident Response*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-88, *Guidelines for Media Sanitization*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-92, *Guide to Computer Security Log Management*	http://csrc.nist.gov/publications/nistpubs/
NIST, SP 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i*	http://csrc.nist.gov/publications/nistpubs/

Common Enterprise Network-Based IDPSs

Product Line	Vendor	URL
Attack Mitigator	Top Layer Networks	http://www.toplayer.com/content/products/index.jsp
BBX	DeepNines	http://www.deepnines.com/bbx.php
Bro	Vern Paxson	http://bro-ids.org/
Cisco IPS	Cisco Systems	http://www.cisco.com/en/US/products/hw/vpndevc/index.html
Cyclops	e-Cop.net	http://www.e-cop.net/
DefensePro	Radware, Ltd.	http://www.radware.com/content/products/dp/default.asp
Dragon	Enterasys Networks, Inc.	http://www.enterasys.com/products/ids/
eTrust Intrusion Detection	Computer Associates	http://www3.ca.com/solutions/Product.aspx?ID=163
Juniper Networks IDP	Juniper Networks	https://www.juniper.net/products/intrusion/
IntruShield	Network Associates	http://www.mcafee.com/us/enterprise/products/network_intrusion_prevention/index.html
iPolicy	iPolicy Networks	http://www.ipolicynetworks.com/products/ipf.html
Proventia	Internet Security Systems	http://www.iss.net/products/product_sections/Intrusion_Prevention.html
SecureNet	Intrusion	http://www.intrusion.com/
Sentivist	Check Point Software Technologies	http://www.nfr.com/solutions/sentivist-ips.php
Snort	Sourcefire	http://www.snort.org/
Sourcefire	Sourcefire	http://www.sourcefire.com/products/is.html
StoneGate	StoneSoft Corporation	http://www.stonesoft.com/en/products_and_solutions/products/ips/
Strata Guard	StillSecure	http://www.stillsecure.com/strataguard/index.php
Symantec Network Security	Symantec Corporation	http://www.symantec.com/enterprise/products/index.jsp
UnityOne	TippingPoint Technologies	http://www.tippingpoint.com/products_ips.html

Common Enterprise Wireless IDPSs

Product Line	Vendor	URL
AirDefense	AirDefense	http://www.airdefense.net/products/index.php
AirMagnet	AirMagnet	http://www.airmagnet.com/products/
AiroPeek	WildPackets	http://www.wildpackets.com/products/airopeek/overview
BlueSecure	BlueSocket	http://www.bluesocket.com/products/centralized_intrusion.html
Highwall	Highwall Technologies	http://www.highwalltech.com/products.cfm
Red-Detect	Red-M	http://www.red-m.com/products-and-services/red-detect.html
RFprotect	Network Chemistry	http://networkchemistry.com/products/
SpectraGuard	AirTight Networks	http://www.airtightnetworks.net/products/products_overview.html

Common Enterprise NBA Systems

Product Line	Vendor	URL
Arbor Peakflow X	Arbor Networks	http://www.arbornetworks.com/products_x.php
Cisco Guard, Cisco Traffic Anomaly Detector	Cisco Systems	http://www.cisco.com/en/US/products/hw/vpndevc/index.html
GraniteEdge ESP	GraniteEdge Networks	http://www.graniteedgenetworks.com/products
OrcaFlow	Cetacea Networks	http://www.orcaflow.ca/features-overview.php
Profiler	Mazu	http://www.mazunetworks.com/products/index.php
Proventia Network Anomaly Detection System (ADS)	Internet Security Systems	http://www.iss.net/products/Proventia_Network_Anomaly_Detection_System/product_main_page.html
QRadar	Q1 Labs	http://www.q1labs.com/content.php?id=175
StealthWatch	Lancope	http://www.lancope.com/products/

Common Enterprise Host-Based IDPSs

Product Line	Vendor	URL
BlackIce	Internet Security Systems	http://www.iss.net/products/product_sections/Server_Protection.html http://www.iss.net/products/product_sections/Desktop_Protection.html
Blink	eEye Digital Security	http://www.eeye.com/html/products/blink/index.html
Cisco Security Agent	Cisco Systems	http://www.cisco.com/en/US/products/sw/secursw/ps5057/index.html
Deep Security	Third Brigade	http://www.thirdbrigade.com/
DefenseWall HIPS	SoftSphere Technologies	http://www.softsphere.com/programs/
Intrusion SecureHost	Intrusion	http://www.intrusion.com/
McAfee Host Intrusion Prevention	McAfee	http://www.mcafee.com/us/enterprise/products/host_intrusion_prevention/index.html
Primary Response	Sana Security	http://www.sanasecurity.com/products/pr/index.php
Proventia	Internet Security Systems	http://www.iss.net/products/product_sections/Server_Protection.html http://www.iss.net/products/product_sections/Desktop_Protection.html
RealSecure	Internet Security Systems	http://www.iss.net/products/product_sections/Server_Protection.html http://www.iss.net/products/product_sections/Desktop_Protection.html
SecureIIS Web Server Protection	eEye Digital Security	http://www.eeye.com/html/products/secureiis/index.html
Symantec Critical System Protection	Symantec	http://www.symantec.com/enterprise/products/index.jsp

This page has been left blank intentionally.

Appendix D—Index

U

V

W