International Journal of Computer Networks and Communications Security

VOL. 3, NO. 10, OCTOBER 2015, 384–395
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Intrusion Detection and Prevention System: Technologies and Challenges

M.Azhagiri[1], Dr A.Rajesh[2] and Dr S.Karthik[3]

[1] Research Scholar, St.Peter's University, Avadi, Chennai-600054, India.

[2] Professor/CSE, C Abdul Hakeem College of Engineering and Technology, Melvisharam, Tamil Nadu 632509, India.

[3] Associate Professor/IT, V.M.K.V Engineering College, Salem, Tamil Nadu 636308, India.

E-mail: [1]azhagiri1687@gmail.com, [2]amrajesh73@gmail.com, [3]mailmecarthic@gmail.com

ABSTRACT

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 detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of IDS and can also attempt to stop possible incidents. This paper provides an overview of IDPS technologies. It explains the key functions that IDPS technologies perform and the detection methodologies that they use. Next, it highlights the most important characteristics of each of the major classes of IDPS technologies. The paper also discusses various types of IDPS security capabilities, technology limitations and challenges.

Keywords: *Intrusion, IDPS Detection, Security.*

1 INTRODUCTION

Intrusion detection is the process of an examining the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or about to happen threats of violation of computer security policies, acceptable use policies, or standard security practices [1]. An intrusion detection system (IDS) is software that automates the intrusion detection process [2]. 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 [3].

IDPSs are primarily focused on detecting possible incidents. For example, an IDPS could detect when an attacker has successfully compromised a system by exploiting vulnerability in the system [4]. 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.

Figure.1 shows an example of an IDS system. The IDPS could also log information that could be used by the incident handlers. Many IDPSs can also be configured to recognize violations of security policies [5]. For example, some IDPSs can be configured with firewall rule set-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 [6].

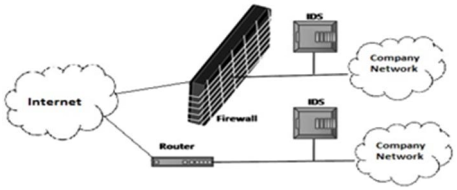

Figure 1 Example of an IDS system

The IDPS could also log information that could be used by the incident handlers. Many IDPSs can also be configured to recognize violations of security policies [5]. For example, some IDPSs can be configured with firewall rule set-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 [6].

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:

i. 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.

ii. 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.

iii. 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.

2 IDPS DETECTION METHODOLOGIES

IDPS technologies use many methodologies to detect attacks. The primary methodologies are signature-based, anomaly-based, and stateful protocol analysis. These methodologies are described in detail below.

2.1 Signature-Based Detection

A signature-based IDS (also known as a knowledge-based IDS) examines data traffic in search of patterns that match known signatures—that is, preconfigured, predetermined attack patterns. A signature is a pattern that corresponds to a known attack or type of attack. Signature-based detection is the process of comparing signatures against observed events to identify possible attacks. Examples of signatures are:

- A telnet attempt with a username of "root", which is a violation of an organization's security policy
- An 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 IDS technology is widely used because many attacks have clear and distinct signatures. The problem with the signature-based approach is that, as new attack strategies are identified, the IDS's database of signatures must be continually updated.

2.2 Anomaly-Based Detection

Anomaly-based detection is the process of comparing definitions of normal activity 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.

The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown attacks. For example, suppose that a computer becomes infected

with a new type of malware. The malware could consume the computer's processing resources, send many e-mails, initiate large numbers of network connections and perform other behavior that would be significantly different from the established profiles for the computer.

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.

2.3 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. 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.

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

3 IDPS COMPONENTS ARCHITECTURE

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

3.1 IDPS Components

The typical components in an IDPS solution are as follows [3]:

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

IDPS components can be connected to each other through regular networks or a separate network designed for security software management known as a management network. If a management network is used, each sensor or agent host has an additional network interface known as a management interface that connects to the management network, and the hosts are configured so that they cannot pass any traffic between management interfaces and other network interfaces [4].

3.2 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 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.

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.

3.3 Architecture Design

The first step in IDPS implementation is designing 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 rule sets to allow IDPS components to communicate.

4 IDPS SECURITY CAPABILITIES

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

4.1 Information Gathering Capabilities

Some IDPS technologies offer information gathering capabilities, such as collecting info-rmation 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.

4.2 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.

4.3 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
- 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.

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.

4.4 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.

5 TYPES OF IDPS TECHNOLOGIES

There are many types of IDPS technologies. For the purposes of this chapter, they are divided into the following four groups based on the type of events they monitor and the ways in which they are deployed: Network-based IDPS, Wireless IDPS, Network behavior analysis (NBA) system and Host-based IDPS. This section discusses each of these four groups in more detail. For each group, it gives a general over view and then discusses the IDPS's security capabilities and limitations in detail.

5.1 Network-Based IDPS

A network-based IDPS monitors and analyzes network traffic for particular network segments or devices to identify suspicious activity. Network-based IDPSs are most often deployed at the division between networks. The IDPS network interface cards are placed into promiscuous mode so that they accept all packets that they see, regardless of their intended destinations. Network-based IDPSs typically perform most of their analysis at the application layer, for example, Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Domain Name System (DNS).

5.1.1 Architecture

Network-based IDPS sensors can be deployed in one of two modes: in-line or passive. An in-line sensor is deployed so that the traffic it monitors passes through it. Some in-line sensors are hybrid firewall/IDPS devices. The primary motivation for deploying sensors in-line is to stop attacks by blocking traffic. Figure 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.

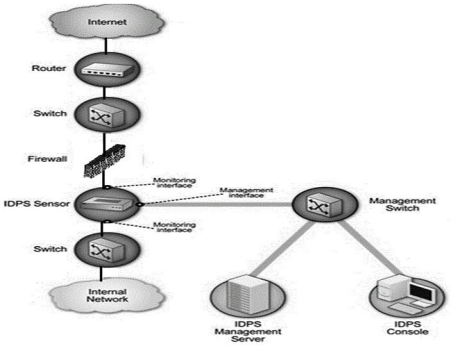

Fig. 2. Inline Network-Based IDPS Sensor Architecture Example

A passive sensor is deployed so that it monitors a copy of the actual traffic; no traffic passes through the sensor. Passive sensors can monitor traffic through various methods, including a switch spanning port, which can see all traffic going through the switch; a network tap, which is a direct connection between a sensor and the physical network medium itself, such as a fiber-optic cable; and an IDS load balancer, which is a device that aggregates and directs traffic to monitoring systems. Most techniques for having a sensor prevent intrusions require that the sensor be deployed in in-line mode. Passive techniques typically provide no reliable way for a sensor to block traffic. In some cases, a passive sensor can place packets onto a network to attempt to disrupt a connection, but such methods are generally less effective than in-line methods [8].

5.1.2 Security Capabilities

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

The types of events most commonly detected by network-based IDPS sensors include application, transport, and network layer reconnaissance and attacks. Many sensors can also detect unexpected application services, such as tunneled protocols, backdoors, and hosts running unauthorized applications. Also, some types of security policy violations can be detected by sensors that allow

administrators to specify the characteristics of activity that should not be permitted, such as TCP or UDP port numbers, IP addresses, and Web site names. Some sensors can also monitor the initial negotiation conducted when establishing encrypted communications to identify client or server software that has known vulnerabilities or is misconfigured. Examples include secure shell (SSH), Transport Layer Security (TLS), and IP Security (IPsec).

Some network-based IDPSs can collect limited information on hosts and their network activity. Examples of this are a list of hosts on the organization's network, the operating system versions and application versions used by these hosts, and general information about network characteristics, such as the number of hops between devices. This information can be used by some IDPSs to improve detection accuracy. For example, an IDPS might allow administrators to specify the IP addresses used by the organization's Web servers, mail servers, and other common types of hosts, and also specify the types of services provided by each host (e.g., the Web server application type and version run by each Web server).

Network-based IDPS sensors offer various prevention capabilities [9]. A passive sensor can attempt to end an existing TCP session by sending TCP reset packets to both end points, to make it appear to each end point that the other is trying to end the connection. However, this technique often cannot be performed in time to stop an attack and can only be used for TCP; other, newer prevention capabilities are more effective. Both passive and in-line sensors can reconfigure other network security devices to block malicious activity or route it elsewhere, and some sensors can run a script or program when certain malicious activity is detected to trigger custom actions.

5.1.3 Technology Limitations

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

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

5.2 Wireless IDPS

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

Most WLANs use the Institute of Electrical and Electronics Engineers (IEEE) 802.11 family of WLAN standards [9.8]. IEEE 802.11 WLANs have two fundamental architectural components: a station, which is a wireless end-point device (e.g., laptop computer, personal digital assistant), and an access point, which logically connects stations with an organization's wired network infrastructure or other network. Some WLANs also use wireless switches, which act as intermediaries between access points and the wired network. A network based on stations and access points is configured in infrastructure mode; a network that does not use an access point, in which stations connect directly to each other, is configured in ad hoc mode. Nearly all organization WLANs use infrastructure mode. Each access point in a WLAN has a name assigned to it called a service set identifier (SSID).The SSID allows stations to distinguish one WLAN from another.

5.2.1 Architecture

The typical components in a wireless IDPS are the same as for a network-based IDPS, other than sensors. Wireless sensors function very differently

because of the complexities of monitoring wireless communications. Unlike a network-based IDPS, which can see all packets on the networks it monitor, a wireless IDPS works by sampling traffic. Figure.3 As with a network-based IDPS, a separate management network or the organization's standard networks can be used for wireless IDPS component communications.

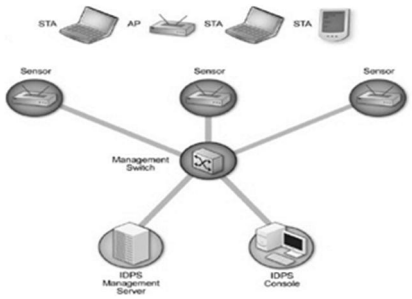

Fig. 3. Wireless IDPS Architecture

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

5.2.2 Security Capabilities

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

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

5.2.3 Technology Limitations

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

Wireless IDPS sensors are also susceptible to attack. The same denial of service attacks (both logical and physical) that attempt to disrupt WLANs can also disrupt sensor functions. Sensors are also often particularly susceptible to physical attack because they are usually located in hallways, conference rooms, and other open areas. Some sensors have anti tamper features, such as being designed to look like fire alarms or regular access points, that can reduce the likelihood that they will be attacked.

All sensors are susceptible to physical attacks such as jamming that disrupt radio-frequency transmissions; 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.

An attacker can passively monitor wireless traffic, which is not detectable by wireless IDPSs. If weak security methods are being used, for example, Wired Equivalent Privacy (WEP), the attacker can then perform off-line processing of the collected traffic to find the encryption key used to provide security for the wireless traffic. With this key the

attacker can decrypt the traffic that was already collected, as well as any other traffic collected from the same WLAN. Wireless IDPSs cannot fully compensate for the use of insecure wireless networking protocols.

5.3 Network Behavior Analysis (NBA) System

An NBA system examines network traffic or traffic statistics to identify unusual traffic flows, such as DDoS attacks, certain forms of malware (e.g., worms, backdoors), and policy violations (e.g., a client system providing network services to other systems). Historically, NBA systems have been known by many names, including network behavior anomaly detection (NBAD) software, network behavior analysis and response software, and network anomaly detection software [10]. NBA solutions usually have sensors and consoles, and some products also offer management servers.

5.3.1 Architecture

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.4 shows an example of NBA network architecture.

In 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.

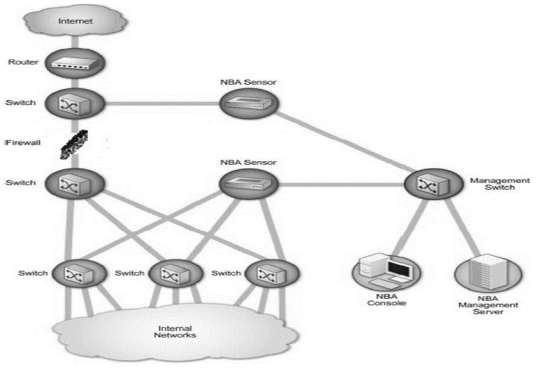

Fig. 4. NBA Sensor Architecture Example

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

5.3.2 Security Capabilities

NBA technologies typically can detect several types of malicious activity. Most products use primarily anomaly-based detection [6], along with some technologies offer no signature-based detection capability, other than allowing administrators to manually set up custom filters that are essentially signatures to detect or stop specific attacks. The types of events most commonly detected by NBA sensors include network-based denial of service attacks, network scanning, worms, the use of unexpected application services, and policy violations (e.g., a host attempting to contact another host with which it has no legitimate reason to communicate). Most NBA sensors can reconstruct a series of observed events to determine the origin of an attack. For example, if worms infect a network, NBA sensors can analyze the worm's flows and find the host on the organization's network that first transmitted the worm.

NBA technologies rely primarily on observing network traffic and developing baselines of expected flows and inventories of host characteristics. NBA products automatically update their baselines on an ongoing basis. As a result, typically there is not much tuning or customization to be done. Administrators might adjust thresholds periodically (e.g., how much additional bandwidth usage should trigger an alert) to take into account changes to the environment. A few NBA products offer limited signature-based detection capabilities. The supported signatures tend to be very simple, primarily looking for particular values in certain IP, TCP, UDP, or ICMP header fields. This capability is most helpful for in-line NBA sensors because they can use the signatures to find and block attacks that a firewall or router might not be capable of blocking. However, even without a signature capability, an in-line NBA sensor might be able to

detect and block the attack because of its flow patterns.

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

5.3.3 Technology Limitations

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

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

5.4 Host-Based IDPS

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

5.4.1 Architecture

Figure.5 shows an example of a host-based IDPS deployment architecture. Host-based IDPS agents are most commonly deployed to critical hosts, such as publicly accessible servers and servers containing sensitive information, although they can be deployed to other types of hosts as well. Some organizations use agents primarily to analyze activity that cannot be monitored by other security controls. For example, network-based IDPS sensors cannot analyze the activity within encrypted network communications, but host-based IDPS agents installed on endpoints can see the unencrypted activity. The network architecture for host-based IDPS deployments is typically simple. Since the agents are deployed on existing hosts on the organization's networks, the components usually communicate over those networks instead of using a separate management network.

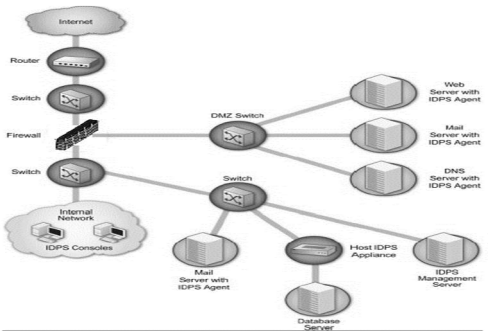

Fig. 5. Host-Based IDPS Agent Deployment Architecture Example

To provide intrusion prevention capabilities, most IDPS agents alter the internal architecture of hosts. This is typically done through a shim, which is a layer of code placed between existing layers of code. A shim intercepts data at a point where it would normally be passed from one piece of code to another. The shim can then analyze the data and determine whether or not it should be allowed or denied. Host-based IDPS agents may use shims for several types of resources, including network traffic, file system activity, system calls, Windows registry activity, and common applications (e.g., email, Web).

5.4.2 Security Capabilities

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

Host-based IDPSs often have extensive knowledge of hosts' characteristics and configurations, an agent can often determine whether an attack would succeed if not stopped. Agents can use this knowledge to select prevention actions and to prioritize alerts. Like any other IDPS technology, host-based IDPSs often cause false positives and false negatives. However, the accuracy of detection is more challenging for host-based IDPSs because they detect events but do not have knowledge of the context under which the events occurred. For example, a new application may be installed – this could be done by malicious activity or done as part of normal host operation. The event's benign or malicious nature cannot be determined without additional context.

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

5.4.3 Technology Limitations

Host-based IDPSs have some significant limitations. Although agents generate alerts on a real-time basis for most detection techniques, some techniques are used periodically to identify events that have already happened. Such techniques might only be applied hourly or even just a few times a day, causing significant delay in identifying certain events. Also, many host-based IDPSs are intended to forward their alert data to the management servers on a periodic basis, such as every 15–60 min, to re duce overhead. This can cause delays in initiating response actions, which especially increases the impact of incidents that spread quickly, such as malware infestations. Host-based IDPSs can consume considerable resources on the hosts that they protect, particularly if they use several detection techniques and shims. Host-based IDPSs can also cause conflicts with existing security controls, such as personal firewalls, particularly if those controls also use shims to intercept host activity.

6 CONCLUSION

Information security has become a legitimate concern for both organizations and computer users due to the growing confidence with computers and electronic transactions. Different techniques are used to support the security of an organization against threats or attacks. On the other side, attackers are discovering new techniques and ways to break these security policies. The four primary types of IDPS technologies – network-based, wireless, NBA, and host-based – each offer fundamentally different capabilities. Each technology type offers benefits over the other, such as detecting some attacks that the others cannot, detecting some attacks more accurately, and functioning without significantly impacting the protected hosts' performance. Accordingly, using multiple types of IDPS technologies can achieve more comprehensive and accurate detection and prevention of malicious activity.

7 REFERENCES

[1] Abdullah A. Mohamed, "Design Intrusion Detection System Based On Image Block Matching", International Journal of Computer and Communication Engineering, IACSIT Press, Vol. 2, No. 5, September 2013.

[2] Denning, Dorothy E., "An Intrusion Detection Model," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131

[3] Karen Scarfone & Peter Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", NIST Special Publication on Computer security, February 2007.

[4] Lunt, Teresa F., "Detecting Intruders in Computer Systems," 1993 Conference on Auditing and Computer Technology, SRI International

[5] Sebring, Michael M., and Whitehurst, R. Alan., "Expert Systems in Intrusion Detection: A Case Study," The 11th National Computer Security Conference, October, 1988

[6] Smaha, Stephen E., "Haystack: An Intrusion Detection System," The Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December, 1988

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

[8] S. Northcutt, L. Zeltser, S. Winters, K. Kent, R. Ritchey: Inside Network Perimeter Security, 2nd edn. (Sams Publishing, Indianapolis 2005)

[9] M. Rash, A. Orebaugh, G. Clark, B. Pinkard, J. Babbin: Intrusion Prevention and Active Response_ Deploying Network and Host IPS (Syngress, Rockland, Massachusetts 2005)

[10] D. Marchette: Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint (Springer, New York 2001)