International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:2, No:10, 2008

Network Based Intrusion Detection and Prevention Systems in IP-Level Security Protocols

R. Kabila

***Abstract*—**IPsec has now become a standard information security technology throughout the Internet society. It provides a well-defined architecture that takes into account confidentiality, authentication, integrity, secure key exchange and protection mechanism against replay attack also. For the connectionless security services on packet basis, IETF IPsec Working Group has standardized two extension headers (AH&ESP), key exchange and authentication protocols. It is also working on lightweight key exchange protocol and MIB's for security management. IPsec technology has been implemented on various platforms in IPv4 and IPv6, gradually replacing old application-specific security mechanisms. IPv4 and IPv6 are not directly compatible, so programs and systems designed to one standard can not communicate with those designed to the other. We propose the design and implementation of controlled Internet security system, which is IPsec-based Internet information security system in IPv4/IPv6 network and also we show the data of performance measurement. With the features like improved scalability and routing, security, ease-of-configuration, and higher performance of IPv6, the controlled Internet security system provides consistent security policy and integrated security management on IPsec-based Internet security system.

***Keywords*—**IDS, IPS, IP-Sec, IPv6, IPv4, VPN.

I. Introduction

IN closer future a new version of the Internet protocol IPv6 should replace an old IPv4 protocol.IPv6 brings many improvements considering simplicity, routing speed, quality of service and security. A new network layer protocol enables flexible use of extended headers, as the IPSec protocol. In spite of these improvements, it is still necessary to take care of network security [7][11].

By transition from IPv4 to IPv6 we use coexistence mechanisms. *Dual stack* is one of the simplest methods for introducing the IPv6 in Internet [1]. Dual stack protocol maintains both IPv4/IPv6 addresses and can communicate with all IPv4/IPv6 network nodes. The second transition mechanism is traffic encapsulation.

In the first stage of IPv6 implementation this mechanism enables encapsulation of IPv6 traffic through the IPv4 Internet. In an advanced stage of the transition process the encapsulation will interconnect the remained IPv4 networks over the IPv6 Internet.

Author is with Amrita School of Engineering, Faculty of Computer Science and Engineering, Bangalore, India (e-mail: kabilacse@gmail.com).

The security issues are especially emphasized in a period of coexistence of IPv4 and IPv6, while the transition mechanisms open new and till now unknown possibilities of an unauthorized invasion. Usually, IPv6 traffic is tunneled through the IPv4 network without conscious of the network administrator. It is therefore extremely important to take care of network security, and to properly implement the protecting mechanisms, as firewalls and the IDS – Intrusion Detection System. IPSec protocol, which is an integral part of IPv6, enables new amazing possibilities for insuring privacy, integrity and authentication of communication. Despite some drawbacks, the IPSec protocol integrated in IPv6 provides an efficient solution for VPN – Virtual Private Networks, *end-to-end* connections.

II. IPv6 Firewalls and IDS Systems

A. IPv6 Firewalls

Firewalls are usually implemented between a LAN and the other insecure networks. They have to check all traffic entering or leaving the network by using different filtering rules, and then leak or block the packets A Linux firewall is based on analyzing a packet header, and results are then compared with a predefined set of rules [6].

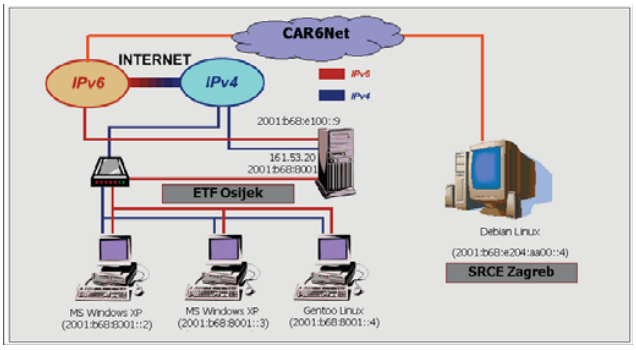

Fig. 1 Experimental IPv6 network

The rules are sequentially performed on each packet, and depending on results, the packet could be discarded, accepted or redirected to an additional analysis. A NetFilter is a tool set in Linux by which filtering rules could be defined. The Netfilter contains an IP Tables tool, which is used for analysis of network packets and execution of defined commands over these packets. All recent Linux distributions (Mandrake, Debian, Red Hat) support IP6 Tables. IP6 Tables contains all

facilities of IP Tables (IP Chains too), and is much faster, more reliable and provides improved network security. A consequence is a much more complex configuration of the IP6 Tables tool. Furthermore, ip6tables software tool for writing the filtering rules in the filtering tables is associated with the IP6 Tables [10]. ICF - Internet Connection Firewall is built in Windows XP OS. It is not available as a single product, neither for other versions of Windows OS nor for other operating systems. ICF defines a protecting border between a host and the Internet as shown in Fig. 1.

B. IDS - Intrusion Detection System

For an improved protection of our network it is recommended to implement an Intrusion Detection System for detection of unauthenticated invasions [9]. IDS is a hardware or a software system that automatically supervises the events in the host or in the network, and analyses them to find out potential security problems. Two main IDS systems are *Network-Based IDS Systems* and *Host-Based IDS Systems.* Most commercial IDS systems are Network-Based IDS Systems. The Network-Based IDS systems detect attacks by capturing and analyzing network packets. In such a way the Network-Based IDS system protects many hosts simultaneously.

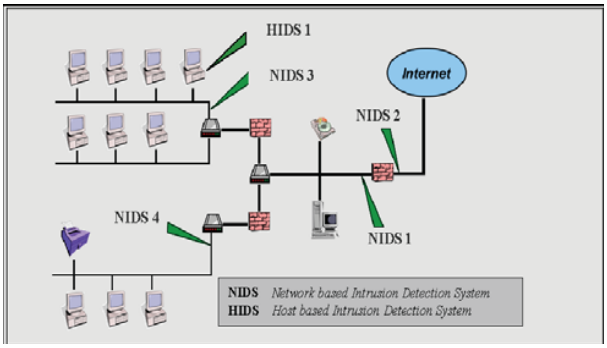

Fig. 2 Positioning of the IDS systems

Fig. 2 shows possible positions for the implementation of IDS systems. The most important position for installing an IDS system is marked by number 1 (the position behind the firewall detaching the LAN and the Internet). This IDS could protect the whole LAN. It is also desirable to put the IDS system in the front of the firewall (position 2), since it could detect all attacks, even those filtered by firewall. If the local network consists of many segments (subnets), it is desirable to position the firewall and the IDS system at every segment (positions 3 and 4 in Fig. 2.). In such a way we could detect unauthorized activities of local users (users authorized inside the local network). Finally, it is desirable to position Host-Based IDS system at every host inside the LAN (position 5). A support for IPv6 protocol and IDS systems is till now (January 2005) very poor. There do not exist official releases of freeware IPv6 IDS tolls.

III. IPV6 IPS Systems

A. Network Based IDPS

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 Sections. 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 [12].

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 container by the layer below it. The four TCP/IP layers, from highest to lowest, are shown in Fig. 3

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.

Fig. 3 TCP/IP Layer

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 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. [16]

2) 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:

3) 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.

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

4) 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.

B. 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 [21].

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:

1) 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. Fig. 4 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.

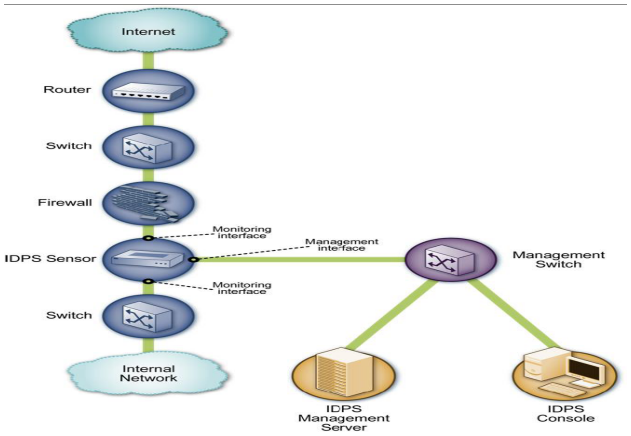

Fig. 4 Inline Network-Based IDPS Sensor Architecture Example

2) 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:

3) 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 see 56-bit subkeys is stored as a 64-bit (eight octet) quantity, with the least significant bit of each octet used as a parity bit 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.

4) 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.

5) 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: [17]

i. 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). Fig. 5 shows examples of passive sensors connected to the monitored network using IDS load balancers, network taps, and spanning ports.

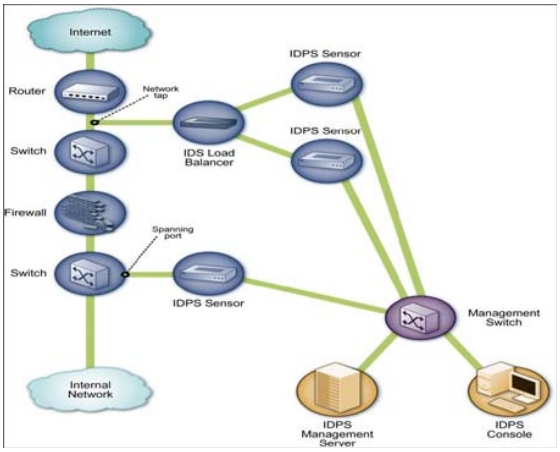

Fig. 5 Passive Network-Based IDPS Sensor Architecture Example

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.

ii. 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.

iii. 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.

IV. Security Capabilities

Network-based IDPS products provide a wide variety of security capabilities. Sections IV through VI 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. [18]

A. 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:

1) 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.

2) 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.

3) 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.

4) 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.

B. 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: [13], [14]

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 type21, 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 request State-related information (e.g., authenticated username 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.

V. 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.

A. Detection Events Types

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

1) Application Layer Reconnaissance and Attacks

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[20].

2) Transport Layer Reconnaissance and Attacks

The most frequently analyzed transport layer protocols are TCP and UDP. (e.g., port scanning, unusual packet fragmentation, SYN floods).

3) Network Layer Reconnaissance and Attacks

The level of IPv6 analysis that network-based IDPSs can perform varies considerably among products. Some 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.

B. Detection Accuracy

Newer technologies use a combination of detection methods to increase accuracy and the breadth of detection. Network-based IDPSs would be able to interpret all network activity just as the endpoints do[2],[3]. 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.

C. Tuning and Customization

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. It 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[7],[8].

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.

VI. Prevention Capabilities

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

A. Passive Only

1) 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*. 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 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. [19]

B. Inline Only

1) Performing Inline Firewalling

Most inline IDPS sensors offer firewall capabilities that can be used to drop or reject suspicious network activity.

2) 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.

3) Altering Malicious Content

As described in Section II 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.

C. Both Passive and Inline

1) 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.

2) 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.

VII. 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.

A. 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 explains the functional areas [15]

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).

2) 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.

3) 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[4],[5]*. 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.

B. 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 updatethe 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 setup 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 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 typicallyused for remote management of sensors or agents and management servers, CLIs are typically used forlocal management of those components. Sometimes a CLI can be reached remotely through an encryptedconnection established through secure shell (SSH) or other means. Consoles are typically much easier touse than CLIs, and CLIs often provide only some of the functionality that consoles provide.

VIII. Summary

A network-based IDPS in IPv6 network monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity. 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 in IP-Sec 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.

Network-based IDPSs IP-sec 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.

References

[1] RFC 1933: Transition Mechanisms for IPv6 Hosts and Routers
[2] RFC 2529: Transmission of IPv6 preko IPv4 Domains without Explicit Tunnels
[3] RFC 1853: IP in IP Tunneling
[4] RFC 3056: Connection of IPv6 Domains via IPv4 Clouds
[5] RFC 2402: IP Authentication Header (AH)
[6] RFC 2406: IP Encapsualtion Security Payload (ESP)
[7] RFC2460: Internet Protocol, Version 6 (IPv6) Specification
[8] RFC 2409 The Internet Key Exchange (IKE)
[9] RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP).
[10] Bill McCarty, Red Hat Linux Firewalls, Wiley Publishing, Indianapolis, Indiana, 2003
[11] N. Sklavos, and O. Koufopavlou, Mobile Communications World: Security Implementations Aspects-A State of the Art, World: Security Implementations Aspects-A State of the Art, CSJM Journal, Institute of Mathematics and Computer Science,
[12] Bace, Rebecca, Intrusion Detection, Macmillan Technical Publishing, 2000.
[13] Bejtlich, Richard, Extrusion Detection, Addison-Wesley, 2005.
[14] Bejtlich, Richard, The Tao of Network Security Monitoring: Beyond Intrusion Detection,Addison-Wesley, 2004.
[15] Crothers, Tim, Implementing Intrusion Detection Systems: A Hands-On Guide for Securing the Network, 2002.
[16] Endorf, Carl et al, Intrusion Detection and Prevention, McGraw-Hill Osborne Media, 2003.
[17] Kruegel, Chris et al, Intrusion Detection and Correlation: Challenges and Solutions,Springer,2004.
[18] Nazario, Jose, Defense and Detection Strategies against Internet Worms, Artech House Publishers, 2003.
[19] Northcutt, Stephen and Novak, Judy, Network Intrusion Detection: An Analyst's Handbook,Third Edition, New Riders, 2003
[20] Rash, Michael et al, Intrusion Prevention and Active Response: Deployment Network and Host IPS, Syngress, 2005.
[21] K. Wang and S.J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," Proc. Seventh Int'l Symp. Recent Advanced in Intrusion Detection (RAID), Sept. 2004.