Intrusion Detection and Prevention System (IDPS) Technology- Network Behavior Analysis System (NBAS)

Tiwari Nitin, Solanki Rajdeep Singh and Pandya Gajaraj Singh
Singhania University, Jhunjhunu, Rajasthan, INDIA

Available online at: www.isca.in
Received 11th July 2012, revised 19th July 2012, accepted 20th July 2012

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 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. In this paper we discuss the one technology of IDPS named network behavior analysis system. A network behavior analysis system (NBAS) is basically an IDPS (intrusion detection and prevention system) technology 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, In this paper we 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 part discusses the management capabilities of the technologies, including recommendations for implementation and operation.

Keywords: Intrusion detection and prevention system (IDPS), network behavior analysis system (NBAS), TCP, UDP, ICMP, time to leave (TTL).

Introduction

An intrusion detection system (IDS)[1] 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. 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.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs[2] 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. The types of IDPS technologies are differentiated primarily by the types 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

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.

In this paper we discuss about 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).

Components of 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 and sFlow. Typical flow data particularly relevant to intrusion detection and prevention includes the following: i. Source and destination IP addresses, ii. Source and destination TCP or UDP ports or ICMP types and codes, iii. Number of packets and number of bytes transmitted in the session, iv. Timestamps for the start and end of the session.

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-1 shows an example of an NBA network architecture.

Methodology

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.

Security Capabilities: NBA products provide a variety of security capabilities. We describe common security capabilities[3], divided into four categories: information gathering, logging, detection, and prevention, respectively.

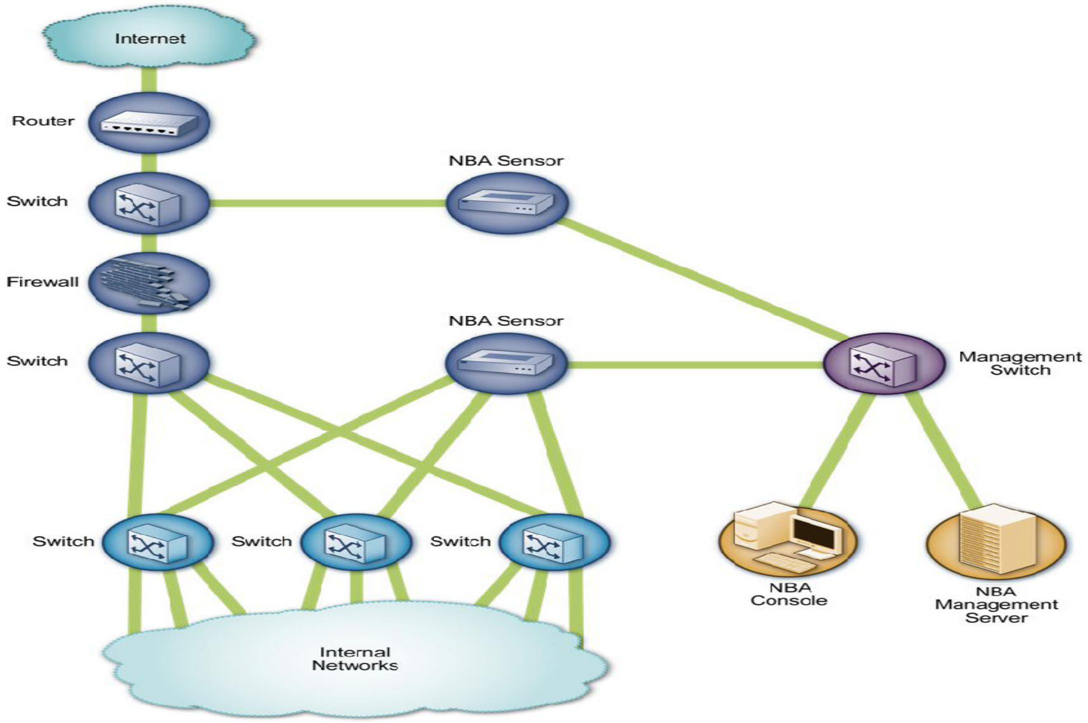

Figure-1
Passive Network-Based IDPS Sensor Architecture Example

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. 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.

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: i. Timestamp (usually date and time), ii. Event or alert type, iii. Rating (e.g., priority, severity, impact, confidence), iv. Network, transport, and application layer protocols, Source and destination IP addresses[4], v. Source and destination TCP or UDP ports, or ICMP types and codes, vi. Additional packet header fields (e.g., IP time-to-live [TTL]), vii. Number of bytes and packets sent by the source and destination hosts for the connection, viii. 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.

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[5], other than allowing administrators to manually set up custom filters that are essentially signatures to detect or stop specific threats. Here we are discussing the following aspects of NBA software detection capabilities: Types of events detected, detection accuracy, tuning and customization, technology limitations.

Results and Discussion

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.

Detection Accuracy: Because NBA sensors work primarily by detecting[6] 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.

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 rule set-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 white list 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.

Technology Limitations: NBA technologies offer strong detection capabilities for certain types of threats, but they also 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; 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.

Prevention Capabilities: NBA sensors offer various intrusion prevention[7] 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.

Management: Most NBA products offer similar management capabilities. Here we are discuss major aspects of management—implementation, operation, and maintenance—and provides recommendations for performing them effectively and efficiently.

Implementation: Once an NBA product has been selected, the administrators need to design architecture, perform NBA component testing, secure the NBA components, and then deploy them. 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.

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[8] 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.

Conclusion

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.

References

1. Scarefone Karen and Mell Peter, Computer Securiy, National Institute of Standard Technology, **(2007)**

2. Networks Security Essentials: Application and Standards by W. Stallings, Pearson Education **(2007)**

3. Shukla Brahma Dutta and Gupta V.K., Performance Interoperability between RDBs and OODBs, *Res. J. Recent Sci.,* **1**, 419-421 (**2012**)

4. Gupta Dhiraj, Shukla Brahma Dutta Constraint of Secured Database in Distributed Database management System, *advancement in computational technique & application*, **1**, 190-194 **(2011)**

5. Sheetlani Jitendra and Gupta V.K., Concurrency Issues of Distributed Advance Transaction Process**,** *Res. J. Recent Sci.,* **1,** 426-429 (**2012**)

6. Gligor V.D. and Shattuck S.H., Deadlock detection in distributed systems, *IEEE Trans. Softw. Eng.* SE-6, **5**, 435-440 **(1980)**

7. Gupta Dhiraj and Gupta V.K., Approaches for Deadlock Detection and Deadlock Prevention for Distributed systems*, Res. J. Recent Sci.,* **1**, 422-425 (**2012**)

8. Mell Peter and Scarfone Karen, Guide to Intrusion Detection and Prevention Systems, U.S. Department of Commerce **(2007)**

9. Moss E.B., Nested transactions: An approach to reliable distributed computing, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA **(1981)**

10. Gray J.N., Notes on database operating systems. In Operating Systems: An Advanced Course, Springer-Verlag, New York, **60**, 393-481 **(1978)**