Performance of the Network Intrusion Detection Systems

M.V.Ramana Murthy, P.Ram Kumar, E.Devender Rao, A C Sharma, S.Rajender ,S.Rambabu

Department of CSE, University College of Engineering, Osmania University, Hyderabad, 500007, India

Abstract
Security is an important factor of the Network Protection. Zero-day attacks, new (anamolous) attacks exploiting previously unknown system vulnerabilities, have become potentially serious threats to the very existence of the Network itself . Defending against them is no easy task. However, having identified "degree of system knowledge" as one difference between legitimate and illegitimate users, theorists have drawn on information theory as a basis for intrusion detection. Intrusion detection systems (IDS) have become one of the most common countermeasures in the network security arsenal. But while other technologies such as firewalls and anti-virus provide proactive protection, most current IDSs are passive by nature. Most current network intrusion detection systems (NIDSs) employ either misuse detection or anomaly detection. However, misuse detection cannot detect unknown intrusions, and anomaly detection usually has high false positive rate. To overcome the limitations of both techniques, we incorporate both anomaly and misuse detection into the NIDS. The proposed approach can improve the detection performance of the NIDSs, where only anomaly or misuse detection technique is used.

Key words:
IDS, Performance, Network Protection.

1. Introduction

A “network" has been defined as “Set of interlinked path for transmission of information resembling a net or a spider web” . This definition suits our purpose well: a computer network is simply a system of interconnected computers. How they're connected is irrelevant.
Network security consists of the provisions made in an underlying computer network infrastructure, policies adopted by the network administrator to protect the network and the network-accessible resources from unauthorized access and the effectiveness (or lack) of these measures combined together[1].
Network security starts from authenticating any user. Once authenticated, firewall enforces access policies such as what services are allowed to be accessed by the network users.[1] Though effective to prevent unauthorized access, this component fails to check potentially harmful contents such as computer worms being transmitted over the network. An intrusion prevention system (IPS)[2] helps to detect and prevent such malware. IPS also monitors for suspicious network traffic for contents, volume and anomalies to protect the network from attacks such as denial of service. Communication between two hosts using the network could be encrypted to maintain privacy. Individual events occurring on the network could be tracked for audit purposes and for a later high level analysis.
Honeypots, essentially decoy network-accessible resources, could be deployed in a network as surveillance and early-warning tools. Techniques used by the attackers that attempt to compromise these decoy resources are studied during and after an attack to keep an eye on new exploitation techniques. Such analysis could be used to further tighten security of the actual network being protected by the honeypot.
DoS (Denial-of-Service) attacks are probably the nastiest, and most difficult to address. These are the nastiest, because they're very easy to launch, difficult (sometimes impossible) to track, and it isn't easy to refuse the requests of the attacker, without also refusing legitimate requests for service.
The premise of a DoS attack is simple: send more requests to the machine than it can handle. There are toolkits available in the underground community that make this a simple matter of running a program and telling it which host to blast with requests. The attacker's program simply makes a connection on some service port, perhaps forging the packet's header information that says where the packet came from, and then dropping the connection. If the host is able to answer 20 requests per second, and the attacker is sending 50 per second, obviously the host will be unable to service all of the attacker's requests, much less any legitimate requests (hits on the web site running there, for example). Such attacks were fairly common in late 1996 and early 1997, but are now becoming less popular.
Some things that can be done to reduce the risk of being stung by a denial of service attack include

Not running your visible-to-the-world servers at a level too close to capacity

Using packet filtering to prevent obviously forged packets from entering into your network address space.

Obviously forged packets would include those that claim to come from your own hosts, addresses reserved for private networks as defined in RFC 1918 [4], and the *loopback* network (127.0.0.0).

Manuscript received October 5, 2009
Manuscript revised October 20, 2009

Keeping up-to-date on security-related patches for your hosts' operating systems.

“Unauthorized access" is a very high-level term that can refer to a number of different sorts of attacks. The goal of these attacks is to access some resource that your machine should not provide the attacker. For example, a host might be a web server, and should provide anyone with requested web pages. However, that host should not provide command shell access without being sure that the person making such a request is someone who should get it, such as a local administrator.

2. Requirements

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization.

An intrusion detection system (IDS) is software that automates the intrusion detection process. On the other hand, an intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. This section provides an overview of IDS and IPS technologies as a foundation for the rest of the publication. It first explains how IDS and IPS technologies can be used. Next, it describes the key functions that IDS and IPS technologies perform and the detection methodologies that they use. Finally, it provides an overview of the major classes of IDS and IPS technologies.

IDS and IPS technologies offer many of the same capabilities, and administrators can usually disable prevention features in IPS products, causing them to function as IDSs. Accordingly, for brevity the term intrusion detection and prevention systems (*IDPS)* is used throughout the rest of this guide to refer to both IDS and IPS technologies. Any exceptions are specifically noted.

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

Recording information related to observed events. Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management (SIEM) solutions, and enterprise management systems.

Notifying security administrators of important observed events. This notification, known as an *alert*, occurs through any of several methods, including the following: emails, pages, messages on the IDPS user interface, Simple Network Management Protocol (SNMP) traps, syslog messages, and user-defined programs and scripts. A notification message typically includes only basic information regarding an event; administrators need to access the IDPS for additional information.

Producing reports. Reports summarize the monitored events or provide details on particular events of interest.

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

IPS technologies are differentiated from IDS technologies by one characteristic:

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

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

Terminate the network connection or user session that is being used for the attack

- Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute
- Block all access to the targeted host, service, application, or other resource.

The IPS changes the security environment. The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target, and altering a host-based firewall on a target to block incoming attacks. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.

The IPS changes the attack’s content. Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned email to reach its recipient. A

more complex example is an IPS that acts as a proxy and *normalizes* incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain attacks to be discarded as part of the normalization process.

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

Most IDPS technologies also offer features that compensate for the use of common evasion techniques. *Evasion* is modifying the format or timing of malicious activity so that its appearance changes but its effect is the same. Attackers use evasion techniques to try to prevent IDPS technologies from detecting their attacks. For example, an attacker could encode text characters in a particular way, knowing that the target understands the encoding and hoping that any monitoring IDPSs do not. Most IDPS technologies can overcome common evasion techniques by duplicating special processing performed by the targets. If the IDPS can "see" the activity in the same way that the target would, then evasion techniques will generally be unsuccessful at hiding attacks.

3. Implementation of Technology

This section covers the major components of IDPS technologies and explains the architectures typically used for deploying the components. It also provides a high-level description of the security capabilities of the technologies, including the methodologies they use to identify suspicious activity.

Components and Architecture

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

Typical Components

The typical components in an IDPS solution are as follows:

_ **Sensor or Agent:** Sensors and agents monitor and analyze activity. The term *sensor* is typically used for IDPSs that monitor networks, including network-based, wireless, and network behavior analysis technologies. The term *agent* is typically used for host-based IDPS technologies.

_ **Management Server:** A *management server* is a centralized device that receives information from the sensors or agents and manages them.7 Some management servers perform analysis on the event information that the sensors or agents provide and can identify events that the individual sensors or agents cannot. Matching event information from multiple sensors or agents, such as finding events triggered by the same IP address, is known as *correlation.* Management servers are available as both appliance and software-only products. Some small IDPS deployments do not use any management servers, but most IDPS deployments do. In larger IDPS deployments, there are often multiple management servers, and in some cases there are two tiers of management servers.

_ **Database Server:** A *database server* is a repository for event information recorded by sensors, agents, and/or management servers. Many IDPSs provide support for database servers**.**

_ **Console:** A *console* is a program that provides an interface for the IDPS's users and administrators. Console software is typically installed onto standard desktop or laptop computers. Some consoles are used for IDPS administration only, such as configuring sensors or agents and applying software updates, while other consoles are used strictly for monitoring and analysis. Some IDPS consoles provide both administration and monitoring capabilities. 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
(e.g., worm attack or distributed denial of service [DDoS] on the monitored networks).

Disadvantages of using a management network include the additional costs in networking equipment and other hardware (e.g., PCs for the consoles) and the inconvenience

for IDPS users and administrators of using separate computers for IDPS management and monitoring.
If an IDPS is deployed without a separate management network, another way of improving IDPS security is to create a virtual management network using a virtual local area network (VLAN) within the standard networks. Using a VLAN provides protection for IDPS communications, but not as much protection as a separate management network. For example, misconfiguration of the VLAN could lead to the exposure of IDPS data. Another concern is that under adverse conditions, such as DDoS attacks or major malware incidents, the network devices shared by the organization's primary networks and VLAN might become completely saturated, negatively impacting the availability and performance of the IDPS.
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). 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. Conclusion

I have incorporated both anomaly and misuse detection into the NIDS (Network Intrusion and Detection System). The proposed approach can improve the detection performance of the NIDSs, where only anomaly or misuse detection technique is used.
The four primary types of IDPS technologies—network-based, wireless, NBA, and host-based—each offer fundamentally different information gathering, logging, detection, and prevention capabilities. Each technology type offers benefits over the other, such as detecting some events that the others cannot and detecting some events with significantly greater accuracy than the other technologies. Accordingly, organizations should consider using multiple types of IDPS technologies to achieve more comprehensive and accurate detection and prevention of malicious activity. In many environments, a robust IDPS solution cannot be achieved without using multiple types of IDPS technologies. For most environments, a combination of network-based and host-based IDPSs is needed for an effective IDPS solution. Wireless IDPSs may also be needed if the organization determines that its wireless networks need additional monitoring or if the organization wants to ensure that rogue wireless networks are not in use in the organization's facilities. NBA technologies can also be deployed if organizations desire additional detection capabilities for DoS attacks, worms, and other threats that NBAs are particularly good at detecting. Organizations that are planning to use multiple types of IDPS technologies, or even multiple products within a single IDPS technology class, should consider whether or not the IDPS products should be integrated in some way. Direct IDPS integration is most often performed when an organization uses multiple IDPS products from a single vendor, by having a single console that can be used to manage and monitor the multiple products. Some products can also share data, which can speed the analysis process and help users to better prioritize threats. A more limited form of direct IDPS integration is having one IDPS product provide data for another IDPS product, such as a networkbased IDPS providing network flow information to an NBA sensor.
Indirect IDPS integration is usually performed with security information and event management (SIEM) software, which is designed to import information from various security-related logs and correlate events among them. SIEM software complements
IDPSs in several ways, including correlating events logged by different technologies, displaying data from many event sources, and providing supporting information from
other sources to help users verify the accuracy of IDPS alerts. An alternative to using SIEM software for centralized logging is the syslog protocol, which provides a simple standard framework for log generation, storage, and transfer that any IDPS can use if designed to do so. Syslog is very flexible for log sources, because each syslog entry contains a content field into which logging sources can place information in any format.
However, this flexibility makes analysis of the log data challenging. Each IDPS may use many different formats for its log message content, so a robust analysis program would need to be familiar with each format and be able to extract the meaning of the data within the fields of each format. Generally, the use of syslog for centralized collection and analysis of IDPS logs does not provide sufficiently strong analysis capabilities to support incident identification and handling.
In addition to dedicated IDPSs, organizations typically have several other types of technologies that offer some IDPS capabilities and complement, but do not replace, the

primary IDPSs. These include network forensic analysis tools, anti-malware n technologies (antivirus software and antispyware software), and firewalls and routers.

REFERENCES

[1] Andrew S.Tanenbaum Computer Networks, 2nd Edition, Prentice Hall of India.
[2] Roger S.Pressman, Software Engineering, 5th Edition, McGraw-Hill International Edition.
[3] www.google.com
[4] www.wikipedia.org
[5] www.eve.mitre.org/eve.
[6] www.silicondefence.com
[7] www.securityfocus.com
[8] www.nipc.gov/cybernotes.htm
[9] www.xforce.iss.net/alerts/summeries.php
[10] www.occure.org/documents/ids
[11] www.ieft.org
[12] www.snort.org
[13] www.whitechats.ca
[14] www.iss.net
[15] www.cisco.com
[16] www.enterasys.com
[17] www.eurocompton.net
[18] www.onlinesecurityfocus.com
[19] www.nss.co.uk
[20] www.csrc-nist.gov
[21] www.iana.org
[22] www.antd.nist.gov
[23] www.wi-fi.org
[24] www.ieee802.org
[25] www.wve.org
[26] www.sflow.org
[27] www.eicar.org
[28] www.wildlist.org

Penumarthy Ram Kumar is currently Associate Professor in Faculty of Computer Science and Engineering in University College of Engineering, Osmania University. He has received his Ph.D(CSE) in 'Modeling Workflow Automation' in 2007 from Osmania University College of Engineering, M.Tech(CSE) from IIT,Mumbai in 1982 and B.Tech(ECE) from Osmania University College of Engineering in 1980.He taught M.Tech(CSE),B.E(CSE) and MCA(CSE) and his active areas of research include Object Oriented Software Engineering, Simulation and Modeling, Workflow Automation, Grid Computing, Distributed Computing, Database Management Systems, Data Mining, Data Security, Network Security.. He has published numerous papers to his credit in IEEE, , ATTI DELLA FOUNDZIONE, International Journal of Computer Science Network Security; etc., and currently leads Simulation and Modeling team in University College of Engineering, Osmania University.

Mangipudi Venkata Ramana Murthy is currently Professor in Faculty of Mathematics and Computer Science in University College of Science, Osmania University. He has received his Ph.D in Computational Fluid Mechanics in 1986 from Osmania University. He is actively involved in research and successfully supervised 22 students for their Doctorial work in the areas of Computer Science and applied Mathematics. The research areas include Artificial Neural Net works, Net work securities , Digital Image processing of Computer Science besides this he also contributed to Fluid Mechanics of Applied Mathematics. He has several research publications to his credit which has international repute such as IEEE, ATTI DELLA FOUNDZIONE, ASME , JFMR, IJHMT etc.

S.Rajender is currently Professor in Methodist Engineering college and Technology, Osmania University, Abids, Hyderabad. He has received his Ph.D in Applied Mathematics in 2009and M.Sc Mathematics in 1983 in Department of Mathematics,Osmania University, Hyderabad-500007

Sakumuri Rambabu is currently working as Associate Professor in Department of Computer Science and Engineering,University College of Engineering,Osmania University from twenty two years. He is basically an M.Tech(CSE),B.Tech(ECE) from Osmania University and was awarded State special Merit scholarship by Government of Andhra Pradesh in 1980 and National Merit Scholarship by Government of India in 1978. He taught to M.Tech(CSE),.,B.E(CSE) and MCA and my areas Research interest include Computer Graphics,Design and Analysis of Algorithms, Operating Systems, DBMS, Data Structures, Discrete Mathematics,Numerical Methods

A.Chandrasekhar Sarma, Biometric Security consultant eM Biosys. Singapore. Faculty in WASE (Wipro Acadamy of software excellance, BITS (Pilani) Coordination programme

Devender Rao is currently working as Associate Professor in Computer Science at Aurora Institute of Information Technology, Ramanthapur. His research interests are in Distributed Computing and Network Securities.