Integrating Intrusion Detection and Network Management

Xinzhou Qin, Wenke Lee
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
{xinzhou, wenke}@cc.gatech.edu

Lundy Lewis
Aprisma Management Technologies
486 Amherst Street
Nashua, NH 03063, USA
lewis@aprisma.com

João B.D. Cabrera
Scientific Systems Company
500 West Cummings Park, Suite 3000
Woburn, MA 01801, USA
cabrera@ssci.com

Abstract

The problems of detecting and resolving performance in distributed systems have become increasingly important and challenging due to the tremendous growth in network-based services. There is a need for a predictive and proactive approach so that appropriate and timely actions can be taken before service disruptions escalate and become widespread. A network management system (NMS) is responsible for monitoring the performance of a network. An intrusion detection system (IDS) is responsible for detecting and responding to intrusions. The current practice is that NMS and IDS are independent to each other in a network. There is little integration and information sharing between the two. In this paper, we outline an approach to integrate NMS and IDS so that the security capabilities of network management can be enhanced and the performance of IDS can be improved.

Keywords

Network Security Management, Intrusion Detection, Anomaly Detection, Intrusion Detection Agent, MIB II, Distributed Denial of Service Attacks

0-7803-7382-0/02/$17.00 ©2002 IEEE

1 Introduction

The distributed denial-of-service (DDoS) attacks on several major Internet sites, such as YAHOO! and CNN, have highlighted the fact that many computer systems and networks are vulnerable to intrusions. While attacking the computer systems and networks has become a kind of "game" for hackers, it has also become a major threat to our daily life, the operation of companies and the security of the nation. Therefore, protecting the network infrastructure has become a critical challenge.

1.1 Intrusion Detection System

An Intrusion Detection System (IDS) is a security mechanism that can intelligently monitor and accurately detect intrusions to the systems and networks in real time, and can respond to the intrusions timely and effectively. An IDS is complementary to other security mechanisms such as firewall, encryption, authentication and access control, in providing security services for networks and systems.

Intrusion detection has been an active research problem for about two decades. James Anderson published an influential paper [3] in 1980. Another mile-stone paper written by Dorothy Denning [10] in 1987 provides a methodological framework for IDS.

Intrusion detection (ID) can be *host-based* or *network-based*. Host-based ID can protect critical network devices storing sensitive and security information. Intrusions are detected by analyzing operating system and application audit trails, e.g., BSM [21]. Network-based ID monitors activities within a network connection(s) or session(s) and performs the analysis on the network traffic.

The two main analysis techniques in intrusion detection are *misuse detection* and *anomaly detection*. Misuse detection uses the signatures of known attacks, i.e., the patterns of attack behavior or effects, to identify the matched activity as an attack. Anomaly detection uses the established normal profiles to identify any unacceptable deviation as the possible result of an intrusion. Anomaly detection is intended to detect new attacks which do not have known signatures yet. Misuse detection lacks such capability of catching new intrusions. However, it has a relatively high accuracy and low false positive rate compared with anomaly detection. Due to the nature of anomaly detection, some legitimate behaviors may be regarded as intrusions due to their deviation from the normal profile, resulting in "false positives".

There are several influential research IDSs. For example, EMERALD [18] uses statistical techniques for anomaly detection and expert system rules for misuse detection. STAT [14] uses state transition analysis for anomaly detection. The approach is based on that the unauthorized activity can be reflected by a certain sequence of actions. These actions indicate that

the system has been moved from an initial authorized state to a compromised state by the intruder. Bro [17] filters network streams into a series of events, and executes scripts that contain site-specific ID rules.

There are many commercial IDS products. In view of intrusion detection techniques, these products are misuse detection systems.

1.1.1 Issues and Challenges in Current Intrusion Detection

High detection rate and low false positive rate are the goals of all IDSs. However, they are also the most serious challenges to the current ID technologies. One reason is the lack of sufficient and efficient information available to the IDS. Current IDS uses BSM or raw packets for analysis. However, only using these raw data sources is inefficient for the IDS to perform the further analysis. Integrating multiple and diverse sources of information can help an IDS improve the detection efficiency, and it can also help an IDS cross-check the analysis results and improve accuracy. Additionally, many IDSs, especially the commercial products, primarily rely on signature matching techniques, which can only detect those known attacks. But, attacks can evolve and many new attacks have emerged. Therefore, using misuse detection technique alone is not enough. Anomaly detection should play a more important role.

Another challenge to current IDSs is the sophistication of attack strategies and attack tools. Attacks are now desired to evade the detection. For example, multi-staged attacks, such as DDoS, have become one of the most difficult intrusions to pro-actively detect, and they are also one of the most dangerous threats to us. These attacks can be distributed and coordinated by using attack relays to achieve the end-goals. Current IDSs lack of the ability to correlate and analyze the related security events in multiple domains, hence cannot detect such attacks effectively. Global intrusion detection is thus a very desirable feature, and is an important research topic.

1.2 Network Management System (NMS)

1.2.1 Security Network Management

Security management is one of the important components in an NMS and it has become an active and important research field. Generally, security management covers such security aspects as access control, authentication, PKI, etc.. Network management community has also enhanced the security capabilities in the new definition of NMS standards. For example, in SNMPv3, encryption and authentication have been added.

However, current security management systems lack efficient and effective capabilities of analyzing and managing the alarms sent by the IDS. As the IDS has become an important component in the network security infrastructure, such managing deficit makes the security management incomplete. In

the field of network management, fault detection and analysis seem to be applicable for detecting intrusions. But compared with the traditional network faults due to causes such as physical link interruption and network device failure, attacks are more subtle and driven by intelligent adversaries. The nature of attacks has made them more difficult to predict and analyze, thus, traditional fault detection techniques are inappropriate for the intrusion detection.

In an NMS, major data resources for analysis are MIBs. However, counting only on MIB is not enough for intrusion detection. For example, information on network connections or header of IP packets are very important for detecting some attacks. In addition, MIB data alone is not complete for the analysis of attack scenario either.

The current practice is to install IDS and NMS in a network in isolation. The purpose of this research is to study how to integrate the ID and NM in terms of audit data sources, analysis techniques, system architecture and deployment strategies. We believe that by integrating the ID and NM, we can bring major changes to the ID technology, significantly improve the effectiveness of ID and enhance the security management capabilities.

The remainder of the paper is organized as follows. In Section 2, we describe our system architecture and its components. A MIB II-based ID model and performance evaluation are presented in Section 3. Section 4 compares our work with related efforts. Finally, we summarize the paper and outline the future work in Section 5.

2 System Architecture

In today's network environment, as the network topology becomes more complicated and attacking methods are "smarter", it is difficult for an IDS to perform effectively by monitoring and analyzing the network from only one observation point. As an IDS performs passive monitoring instead of active filtering as a Firewall does, the IDS needs to be as efficient as possible. In a high-speed and high-volume traffic network, it is also very easy for a hacker to crash or delay the IDS by overloading it for a period of time.

A better deployment strategy for network intrusion detection is to place a number of light-weight agents, namely ID Agents, to various network components. Our experiments have shown that one IDS cannot cover all possible attacks to the network in an accurate and timely fashion, especially in a high-volume traffic network. We will instead use multiple dedicated ID Agents and each of them specializes in a certain category of intrusions. In our architecture, for example, the host-based ID Agents can analyze BSM audit data, system call traces, or user shell command streams, to monitor applications and user behaviors on that host. As for the network ID Agents, they are responsible for detecting those attacks that exploit the weakness of the network protocols, such as DDoS and probing attacks. The number of ID Agents and

deployment strategy are based on the security policy of the enterprise. For example, we can install host-based ID Agents on the critical servers to protect them from being hacked. The network-based ID Agent can be deployed on the routers or switches to monitor the network traffic running through these devices.

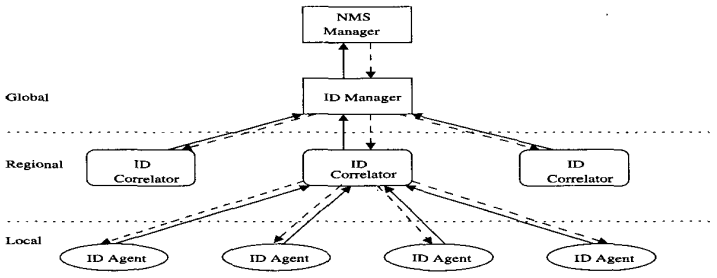

Figure 1: Hierarchical ID Architecture

In addition to the distributed deployment, our system architecture is designed hierarchically. As shown in Figure 1, we divide the protection and analysis scope as *Local_Analysis*, *Regional_Analysis* and *Global_Analysis*. The ID Agents are deployed locally and act the intrusion detection on network components and services. Their coverage scope is usually within the sub-network such as a subnet of a department. Each ID Correlator manages some local ID Agents, combines the security events or alarms sent by local ID Agents, and correlates the intrusion alarms within its domain. ID Correlator is in charge of the whole security activity within its regional coverage and reports to the ID Manager. ID Manager is responsible for the intrusion detection within the whole network such as a campus network. It correlates the ID reports from ID Correlators in multiple domains to make a global intrusion analysis. More complete and complex attack scenario analysis is done by ID Manager. ID Manager sends the final intrusion detection reports to the network manager, which is responsible for the management of the whole network besides the security management. ID Correlator and ID Manager send out intrusion response commands based on security policies to their receivers, i.e., ID Agents and ID Correlators, respectively.

2.1 ID Agent

As discussed above, an ID Agent is responsible for a particular type of intrusions. Figure 2 shows the components of the ID Agent.

The main modules are:

Detection Module: It analyzes the incoming packets, BSM records, MIB values, etc. and matches with the intrusion patterns or the normal pro-

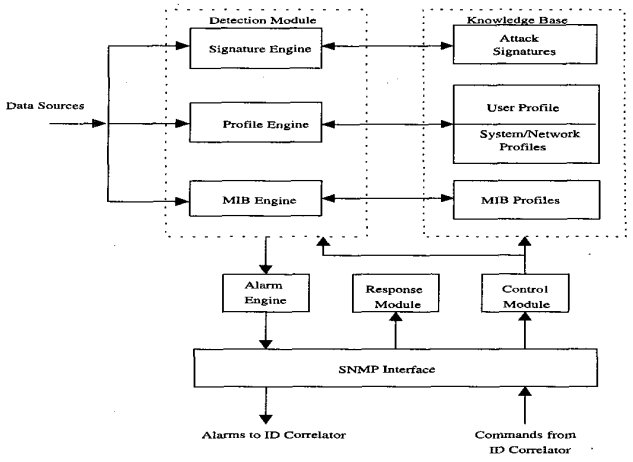

Figure 2: ID Agent

files. There are three engines: *Signature Engine*, *Profile Engine* and *MIB Engine*. Signature Engine focuses on detecting the known intrusions by checking against the attack signatures. Profile engine is responsible for anomaly detection based on the normal profiles of either the network or users. These two engines use "traditional" ID data sources such as BSM or raw packets. MIB Engine checks the related values of MIB objects and compares them with the normal MIB profiles to detect the intrusions. Different ID Agents can contain one or a combination of these three detection engines based on their detection roles. For example, an ID Agent can be equipped with only a Signature Engine in order to detect the well known intrusions fast and efficiently. The combination of these three engines can improve the effectiveness of the ID Agent. The profiles and attack signatures can be modified and updated by the ID Correlator via the module of *Control Module*.

Knowledge Base: It stores the attack signatures, user/system profiles and MIB profiles used by the detection engines. It also stores the rule sets and attack patterns. The Knowledge Base is constructed based on both domain knowledge and results from the analysis tools, such as pattern construction from expert system.

Response Module: It takes corresponding actions such as rejecting connection request from a suspicious source. The corresponding actions are based on the intrusion response commands sent by the ID Correlator.

Control Module: It adjusts the Knowledge Base and the correlation engines based on the control information sent by the ID Correlator. For example, when the topology of user's network has changed, the new topol-

ogy information is sent to the *Control Module* from the ID Correlator to update the Knowledge Base. The attack signatures and rule sets of detection engines will also be updated accordingly.

Alarm Engine: It sends out alarms to the ID Correlator.

SNMP Interface: It is the communication platform between ID Agents and ID Correlator. The alarms from ID Agent and the control information from ID Correlator will be encapsulated into SNMP packets. The advantage is that the whole system can be compatible to the existing NMS. In this approach, alarms are represented by a set of MIB objects.

2.2 ID Correlator

ID Correlator, as shown in Figure 3, is an important component in our architecture. The main responsibilities of the ID Correlator are: 1) to correlate the alarms sent from ID Agents to identify the nature of intrusions, predict the intrusion trend and prevent the potential intrusions; 2) to send alarms to the ID Manager for further intrusion analysis and global correlation; and 3) to respond to intrusions by taking appropriate actions, such as shutting down the compromised network elements, closing the sessions or sending response commands to ID Agents;

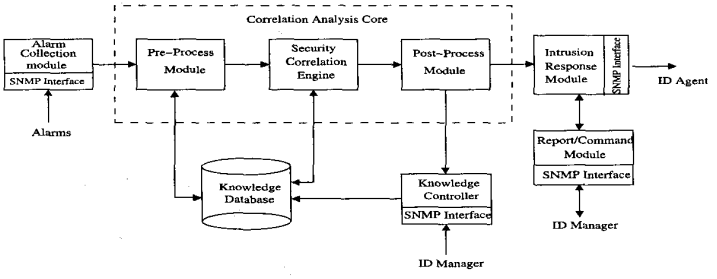

Figure 3: ID Correlator

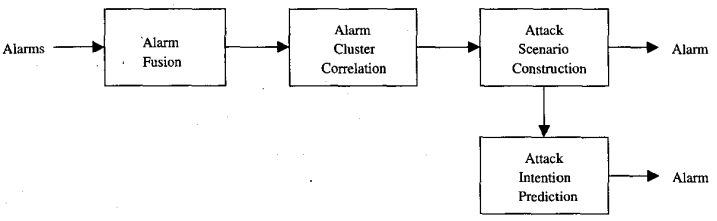

Figure 4: Security Correlation Engine

In the ID Correlator, the main modules are:

Alarm Collection Module: to receive the events/alarms sent by ID Agents.

Security Correlation Engine: to match the alarms sent by ID Agents to correlation patterns, interpret the correlations, predict and identify the intrusion purposes and trend. Besides sending out alarms to the ID Manager, it also has capability of responding to the intrusions by generating the action codes to the ID Agents.

As shown in Figure 4, there are several major components in the Security Correlation Engine:

- *Alarm Fusion*: to correlate and fuse the alarms sent from different ID Agents in order to reduce the number of alerts corresponding to the same intrusion. Alarms are classified into corresponding alarm clusters.
- *Alarm Cluster Correlation*: to correlate the alarms in different categories that indicate the various nature of intrusions in order to reach more accurate and comprehensive attack detections. It also provides information for further attack scenario analysis.
- *Attack Scenario Construction*: to construct the attack scenario based on the previous correlation output. An attack scenario is a sequence of related attack steps. The purpose of the attack scenario construction is to understand what has happened and to predict what will happen next.
- *Attack Intention Prediction*: to recognize the intention of the attacks and to predict what will probably happen next. It provides a pro-active analysis on multi-step intrusions so that proactive actions can take place.

Knowledge Database: to store the knowledge of the network, such as the network topology, network components. Depending on what correlation techniques the correlation engine uses, it will store the related information. For example, if the correlation engine uses rule-based reasoning (RBR) [20], the rule-sets will be stored in the Knowledge Base. If case-based reasoning (CBR) [20] is the core technique in the correlation engine, the Knowledge Base will store the case library that is the knowledge gained from the experience. The knowledge related to the intrusions, such as attack scenarios, intrusion signatures, will also need to be stored in the Knowledge Database. In addition, it stores the the security policy and responses to the intrusions. The response knowledge can be tuned and modified by the ID Manager.

Intrusion Response Module: to send out the intrusion response commands to ID Agents based on the security policy defined for the domain the ID Correlator covers. If there is no existing response knowledge available to the intrusions, the ID Correlator will pass the response duty to the ID Manager, which adds the new response to the Knowledge Database for future use.

Knowledge Controller: to construct, manage and update the Knowledge Database. For example, when a new case occurs, this module will process it and add it to the knowledge base for future use. ID Manager also add or modify the security policies stored in the Knowledge Database via this component.

2.3 Discussion

This distributed and hierarchical system can work well in a high-speed environment because the analysis work has been distributed to the ID Agents.

In our integrated ID and NM environment, we have two levels of correlation. At the lower level, we need to correlate multiple sources of information sent to the ID Agent to determine what the local intrusions are. At the higher level, ID Correlator and ID Manager analyze attack scenario and detect coordinated and distributed attacks. Based on the global correlation, we can answer such questions as what will happen next? When? Where? What can we do to prevent it from happening? The objectives of this hierarchical correlation architecture are to achieve: lower false alarm rate and higher detection rate; intrusion trend prediction and purpose identification, especially for sophisticated and coordinated attacks.

Integrating information from multiple sources has been a key for the IDS to detect the sophisticated attacks that can span multiple hosts, subnets, domains, and can last for a long time period. One of the major factors that contribute to the high false alarm rate is the lack of sufficient information. An IDS that is monitoring and analyzing network packets may see the different information from what the destination hosts observe because different operating systems implement the details of network protocols differently. If network packet data is the only source of available information, the IDS can generate false alarms when it sees "suspicious" traffic that can do no harm in fact. In our architecture, the ID gent correlates the alarms generated by the different detection engines to determine if the anomaly is a true intrusion, and can make a final alarm report to the ID Correlator. Such model that can combine the abnormal information from multiple sources and identify an anomaly is highly desirable.

In order to increase the efficiency of intrusion detection, a MIB detection module in ID Agent is also necessary. For example, for the traffic-based attacks, such as DoS, *traffic counts* are the "right" statistics for detecting intrusions. However, since most of the current IDSs use raw packets for network intrusion detection, the IDS engine needs to calculate the related statistics based on the raw packets in order to detect the abnormal traffic. Such approach is inefficient in terms of space and time. In addition, to select and construct the statistical features to detect a wide range of attacks is very challenging. NMS has already provided us with a comprehensive set of variables

about network activities that are very valuable for anomaly detection. For example, MIB II has many objects representing the traffic information and configuration of the network and hosts. Since the MIBs are already available to whoever installs the SNMP agents, why not use them for intrusion detection? As our experiments show in Section 3.2, by taking advantage of the information from the MIB II, we can do anomaly detection on the traffic-based intrusions directly. In addition, some error statistic based on MIB II objects can also be used for proactive detection. For example, in our previous work [7], object *udpInErrors* is extracted as a key variable at "slave machines", i.e., attacking machines to do proactive detection of DDoS. RMON MIB is also a very good data source for the ID Agents to monitor the operations of the network and applications running on the hosts connected. Therefore, we believe that using the information in NMS can ease the workload of IDS in terms of information collection and calculation.

The intrusion detection information provided by IDS can enhance the NMS's capability of managing and analyzing the operation of the network, especially in the realm of security management.

In the real-world, the network management staff and security staff for ID agent can cooperate, and the integration of ID and NM will make the operations more conveniently and efficiently since such integration will result in the unified interfaces (e.g., reports) and operation standards/policies (e.g., responses). It can therefore improve the ease-of-use, acceptance and effectiveness of both technologies.

3 MIB Detection Engine

In this section, we briefly describe our MIB II-based anomaly detection model and its performance evaluation.

3.1 Model Design Overview

MIB II has several groups such as IP, ICMP and TCP group to collect the information on different layers and protocols. In each group, MIB II variables provide information on system configuration, network traffic, control and error statistics. Therefore, MIB II objects provide us an audit source from which we can have a comprehensive understanding about conditions and operations of the managed elements and network.

In our MIB II based anomaly ID model, we use 91 MIB II objects as our monitoring variables. These objects are related to network traffic, error and configuration information, and are in the groups of IP, ICMP, UDP, TCP and SNMP.

In order to have a comprehensive understanding of network activities, we partition the 91 MIB variables into 18 sub-modules based on their definitions

as well as the layers and protocols that they belong to [19]. The architecture is designed based on the protocol-hierarchy property of MIB II and definitions of the objects.

When applying our anomaly detection model, all sub-modules are used, and if there are any anomalies, the corresponding sub-module will send out the alarms. When an alarm is fired by any of the protocol ID sub-modules, we consider an anomaly in that protocol. In anomaly detection, we do not assume to have prior knowledge about the intrusion, such as what protocol or which layer it will manifest itself. With ID sub-module in each protocol, we can detect an anomaly more precisely and locate the protocol and layer where the anomaly has happened.

When building anomaly detection models, we use classification techniques to generate a rule set for each object to predict its normal values. In our study, we use RIPPER [8], a rule learning algorithm based on *Incremental Reduced Error Pruning* (IREP) as our classification engine because of its good generalization accuracy and concise rule conditions.

We use an object-time approach in constructing the anomaly detection model of each MIB object. Specifically, for each MIB object, we use the previous values of this object to predict the coming value. That is, we use $O_{t-n}, O_{t-(n-1)}, ..., O_{t-2}, O_{t-1}$ to predict the value O_t. The intuition is that there exists a predictable temporal relationship under normal conditions for each object, and such temporal relationship may be broken or disturbed by the abnormal conditions such as the occurrence of intrusions. We train the ID model to learn such normal temporal relationship and detect the potential intrusions.

We use the $Accuracy/Cost$ for selecting the optimal sequence length for each object model. As in [16], we use the running time as the measurement of the cost. Here, we define the time cost as the running time when the test data sets are fed into the model for processing and calculation.

Having determined the optimal sequence length n, for each object, we apply RIPPER to construct rule sets for object model, then establish the normal profiles of sub-ID modules and apply cluster techniques for anomaly detection. The deviation above or below the normal profile by a tolerance deviation margin is regarded as anomaly [19].

3.2 Experiments and Evaluation

In evaluating the ID model, we ran several DDoS attacks using the TFN2K and Trinoo attacking tools [9]. The MIB II variables were collected at the target, and the sample rate and period were the same as that in the normal runs in which no intrusions took place. In this experiment, we set the tolerance deviation margin α as 30%. The margin was selected based on the observation of normal traffic and the security policy.

Type of Intrusions	TP[1]	FP[2]	Detected by ID Sub-Module
Ping Flood	95.48%	0.217%	$IP\_In, ICMP\_In$ $ICMP\_Out$
Syn Flood	96.15%	0.767%	IP_In, TCP_In
Targa3	96.78%	0.375%	IP_In, IP_Other TCP_In_Error UDP_In_Error
UDP Flood	97.95%	0.412%	IP_In, UDP_In UDP_In_Error
Mix Flood	97.85%	0.492%	$ICMP\_In, ICMP\_Out$ TCP_In, UDP_In_Error
TearDrop	60.78%	0.327%	IP_In, IP_Other

[1] TP: True Positive Rate, [2] FP: False Positive Rate

Table 1: Intrusion Detections by MIB II-based ID Model

In the performance evaluation, we use our ID model to identify the anomaly sample points. Table 1 shows the performance results of our ID model in the experiments. It also shows which ID sub-module is used to detect each intrusion. From the table, we can see that our ID model has a very high detection accuracy and relatively low false alarm rate on some intrusions such as Ping Flood, Syn Flood, Targa3, UDP Flood and Mix Flood. It also shows that the right sub-modules were able to detect the intrusions that have influences on the corresponding protocols. For example, in the Ping Flood attack, the attacker sends a huge amount of *icmp echo requests* to flood the victim host. The host will not only receive a large number of *icmp echo requests*, but send out many *icmp echo replies* as well. Therefore, we would expect the sub-modules *ICMP_In* and *ICMP_Out* to detect such anomaly. As shown in the results, these two sub-modules *ICMP_In* and *ICMP_Out* have detected the Ping Flood successfully. In TearDrop, the detection rate is not as high as the others. This is due to the sample rate of the ID model. Generally, when an intrusion occurs in the network, the deviation on the MIB data measures will last for a while due to the fact that the intrusion effects have the "build-up" and "residual" stages. However, the relatively large sample interval with regard to the short duration of TearDrop makes these stages less visible and affects the detection performance.

From our experiments, we can see that our MIB II-based ID model can detect some traffic-based attacks such as DDoS effectively. However, due to the limitation of MIB II objects, our ID model currently cannot detect those non-traffic based attacks such as illegal-remote-root-access attacks since such kind of attacks generally do not depend on the large traffic to compromise the target. However, most of MIB II objects are related to the traffic information.

Although our ID model has some limitations, it has shown some potential benefits. First of all, our ID model can improve the efficiency of the intrusion detection. As discussed in the Section 2.3, MIB II variables have provided us with a set of comprehensive information and statistics about network activities, which are very important for anomaly detection. Since the information is provided by NMS, the IDS does not need to duplicate the work, and is thus more efficient.

Another potential application of our MIB II-based ID model is to proactively detect attacks such as DDoS [7]. We can install ID agents based on our MIB II-based ID model on the slave machines to monitor the anomaly activities on the these attacking hosts. When the the slave machine sends attacking packets, the anomalous outbound attacking traffic can be detected by the ID model, and alarms can be sent out before the attacking traffic reach the breaking point to compromise the target.

4 Related Work

Anomaly detection has been an active research topic in the ID research community. NIDES [2] analyzes audit data of the user behavior to search for the unusual or malicious user activities. It has two subsystems. The anomaly detection subsystem uses statistical techniques to monitor the anomaly in the user's profile. The complementary rule-based subsystem is responsible for detecting the known intrusive activities. The rule sets are generated by expert systems. [23] uses the sequence of system calls in the kernel of the operating system as the data sets to represent the normal behavior and detect anomaly intrusions. Several different methods are applied and compared in their work. Lee [15] applies data mining technique to the audit data for the user anomaly detection. These systems all focus on the host-based anomaly detection with analyzing the system programs and resources. EMERALD [18] is an IDS more focused on the network intrusion detection. It has a hierarchical architecture and the enterprise-wide correlator combines the alarms from the distributed detectors. Its anomaly detection model is based on the statistical technique for monitoring the network traffic. However, this system does not take advantage of MIB objects as an audit source for the network-based anomaly detection.

In the security management, [11] designs a protocol that coordinates the inter-domain security management agents to communicate the information of the security policy defined in each policy domain. [6] uses statistical techniques and Bayesian networks to detect the anomalies occurred in the service elements such as the E-mail service. [5] proposes a security management system for intrusion detection by deploying intelligent agents. However, their system does not take advantage of the existing IDS. In our approach, the ID Agent can be an IDS sensor using the "traditional" data source for intrusion

detection. One the of advantages of this approach is that we do not have to duplicate the work that has been done by the existing IDS. As discussed in this paper, to integrate the ID and NM and to combine the strengths of NM and ID can effectively improve the performance of IDS and enhance the capabilities of security management system.

In the fault detection management, work concentrates more on network performance modeling for the purpose of detecting degradation and faults [12, 13]. However, the context of intrusion detection is more subtle than the general fault detection in the NMS. In [22], MIB II variables are used for the fault detection. The approach uses five MIB II objects in the Interface and IP group and focuses on the faults occurred on the IP layer. However, this approach is inappropriate for the intrusion detection since such a few MIBs are not insufficient to represent the behavior of network activities for the anomaly detection. In addition, only focusing on the fault detection on the IP layer is insufficient for the intrusion detection because it is important and desirable to specify all the compromised protocols and layers when detecting the attacks.

Several leading authorities have recently expressed disbelief for the lack of research on the integration [1, 4] of ID and NM as such doable topic is so realistic and important in the security management. There have been little serious efforts, to the extent and goals described in this paper, in integrating ID and NM.

5 Conclusion

In this paper, we analyzed the challenges in current IDS and network security management. We proposed an approach for the integration of ID and NM. We showed a distributed and hierarchical system that can be integrated with existing NMS. We also proposed an hierarchical correlation architecture for improving the detection accuracy and identifying coordinated intrusions. In summary, we believe that the integration of ID and NM will bring changes to ID technology and significantly improve its effectiveness.

We also briefly introduced an MIB II-based model for anomaly detection. We apply data mining techniques to the model construction and anomaly detection. The experiment results showed that our ID model performs well in detecting some traffic-based intrusions such as DDoS. The promising results show that our ID model can play an important role in the integration of the IDS and NMS.

For further work, we will focus on anomaly detection on intrusions with slower traffic and stealth attacks. We will also refine our algorithms of the global correlation and intrusion prediction as well as the prototype system .

6 Acknowledgments

This research is supported in part by a grant from DARPA (F30602-00-1-0603) and a graduate fellowship from the North Carolina Network Initiatives (with funding from Aprisma Management Technologies).

References

[1] E. Amoroso. *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response*. Intrusion.Net Books, 1999.

[2] D. Anderson, T. Frivold, and A. Valdes. Next-generation intrusion detection expert system (NIDES): A summary. Technical Report SRI-CSL-95-07, Computer Science Laboratory, SRI International, Menlo Park, California, May 1995.

[3] J. P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.

[4] R. Bace. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

[5] K. Boudaoud, H. Labiod, R. Boutaba, and Z. Guessoum. Network security management with intelligent agents. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2000)*, April 2000.

[6] A. Bronstein, J. Das, M. Duro, R. Friedrich, G. Kleyner, M. Mueller, S. Singhal, and I. Cohen. Self-aware services: Using Bayesian networks for detecting anomalies in Internet-based services. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM 2001)*, May 2001.

[7] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra. Proactive detection of distributed denial of service attacks using MIB traffic variables - a feasibility study. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM 2001)*, May 2001.

[8] W. W. Cohen. Fast effective rule induction. In *Machine Learning: the 12th International Conference*, Lake Taho, CA, 1995. Morgan Kaufmann.

[9] P.J. Criscuolo. Distributed denial of service - trin00, tribe flood network, tribe flood network 2000, and stacheldraht. Technical Report CIAC-2319, Department of Energy - CIAC (Computer Incident Advisory Capability), February 2000.

[10] D. Denning. An intrusion detection model. *IEEE Transactions on Software Engineering*, 13(2), February 1987.

[11] Zhi Fu, He Huang, Tsung-Li Wu, S. Felix Wu, Fengmin Gong, Chong Xu, and Ilia Baldine. Iscp: Design and implementation of an inter-domain security management agent coordination protocol. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 200)*, April 2000.

[12] J. L. Hellerstein, F. Zhang, and P. Shahabuddin. An approach to predictive detection for service management. In *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management*, May 1999.

[13] L. L. Ho, D. J. Cavuto, S. Papavassiliou, M. Z. Hasan, F. E. Feather, and A. G. Zawadzki. Adaptive network/service fault detection in transaction-oriented wide area networks. In *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management*, May 1999.

[14] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering*, 21(3):181–199, March 1995.

[15] W. Lee, S. J. Stolfo, and K. W. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, May 1999.

[16] W. Lee and D. Xiang. Information-theroetic measures for anomaly detection. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, May 2001.

[17] V. Paxson. Experiences learned from bro. *;login: The USENIX Association Magazine*, September 1999.

[18] P. A. Porras and P. G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In *National Information Systems Security Conference*, Baltimore MD, October 1997.

[19] X. Qin, W. Lee, L. Lewis, and J. B.D. Cabrera. Using MIB II variables for network intrusion detection. In *S. Jajodia and D. Barbarā, editors, Data Mining for Security Applications, Advances in Computer Security*. Kluwer Academic Press, March 2002.

[20] M. Subramanian. *Network Management: Principles and Practice*. Addison-Wesley, 2000.

[21] SunSoft. *SunSHIELD Basic Security Module Guide*. SunSoft, Mountain View, CA, 1995.

[22] M. Thottan and C. Ji. Proactive anomaly detection using distributed intelligent agents. *IEEE Network, Special Issue on Network Management*, April 1998.

[23] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, May 1999.