A NETWORK MONITOR MODEL FOR INTRUSION DETECTION

by

MARGARET ELIZABETH WILLIAMS TUBESING

B.S., United States Military Academy, 1982

A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

COMPUTER SCIENCE

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Approved by:

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

THE INTRODUCTION

1.0 OVERVIEW

Growing steadily alongside advanced computer technology and information sciences is the need for sophisticated computer security methods. Unfortunately, this need is not expected to diminish, but instead is continually expanding as security experts strive to keep pace with the "bad guys" or intruders, i.e. those who in some way attempt to misuse computer systems. It is a frustrating struggle as the experts develop new hardware and software that place additional locks on the computer systems and the intruders inevitably find ways to pick them.

There is a hopeful approach to this problem - auditing. Computer auditing is a function that is based upon detecting the occurance of predetermined events and then recording appropriate detailed information about these events as they occur in the computer system. Different forms of auditing have been used since the very early days of computing, usually motivated by integrity or accounting needs rather

than by security concerns. Auditing logs consist of audit records which an auditor or security officer must manually, or in some cases with limited automation aids, review for suspicious events or unusual patterns of use. In reviewing the logs, the security officer often attempts to discern the audit trail of a particular user. An audit trail is "a set of records that collectively provide documentary evidence of processing used to aid in tracing from original transactions forward to related records and reports, and/or backwards from records and reports to their component source transactions." [DoD85] These review methods are extremely time-consuming and only marginally effective. Obviously, it would be quite difficult for a reviewer to detect any well-disguised intrusions, e.g. those which developed over time and thus were interspersed among hundreds of benign audit records. A partial solution to this problem may be tools based on automated audit trail analysis. Automated analysis, coupled with auditing techniques which are specific to security related events, when combined with other computer security measures can detect intruders. Such systems are called intrusion detection systems.

Dorothy Denning, one developer of such a system, states four factors which motivate the development of a real-time intrusion-detection system [Denn85]:

1) most existing systems have security flaws that render them susceptible to intrusions, penetrations, and other forms of abuse; finding and fixing all these deficiencies is not feasible for technical and economic reasons;

2) existing systems with known flaws are not easily replaced by systems that are more secure - mainly because the systems have attractive features that are missing in the more-secure systems, or else they cannot be replaced for economic reasons;

3) developing systems that are absolutely secure is extremely difficult, if not generally impossible; and

4) even the most secure systems are vulnerable to abuses by insiders who misuse their privileges.

Thus there is a perceived need for intrusion detection systems (IDS's) to back up security mechanisms and there are some promising research efforts to meet that requirement [Denn82], [Lunt88b]. All efforts to date on IDS have concentrated on single site intrusion; this paper provides a framework for research in intrusion detection systems which operate within a network environment.

The remainder of this introductory chapter is divided into four parts. Section 1.1 provides the reader with an overview of computer security threats and abuses and which of these might be partially detected with auditing techniques. Next, Section 1.2 describes current relevent research in the area of automated audit trail analysis. Section 1.3 discusses the goals of this research and

provides a brief outline of the remaining chapters.

1.1 COMPUTER SECURITY THREATS

In order to develop a system that can identify intrusions into a computer system or that can detect breaches of computer security it is first necessary to define or characterize these security threats. A threat is defined as the potential possibility of a deliberate unauthorized attempt to:

a) access information

b) manipulate information

c) render a system unreliable or unusable [Ande80].

An intrusion implies that someone without authority has gained access to some part of a computer system whether it be the system as a whole, a host, a particular directory, or a particular account. Once this person has unauthorized access to an object, he/she can perform further abuses such as copying or altering information, granting further unauthorized privileges to himself/herself and others or divulging this new found information to other unauthorized parties. Breaches of computer security imply the improper or illegal use of proper authority. An example would be an authorized user who passes out sensitive information to which he/she has authorized access. One way in which this is done is through a covert channel. A covert channel is a

communication channel that allows a process to transfer information in a manner that violates the system's security policy [DoD85]. "Every bit of information in the system (that is, every object) that can be modified by one process and read by another - directly or indirectly - is potentially a covert channel." [Gass88]

> DEFINITION 1.1 SECURITY POLICY. The set of laws, rules and practices that regulate how an organization manages, protects and distributes sensitive information. [DoD85]

Anderson, in some early research on audit trail analysis [Ande80], distinguished between two categories of computer system intruders:

1) External Penetrators - these persons are not authorized to use the computer system. They may be from outside the organization and may not even have physical access to the computer, or they may be of the organization but are not intended to use the system. These penetrators might attempt intrusion by wiretapping or by posing as an authorized user.

2) Internal Penetrators - these are further classified as:

   a. Masqueraders - to the machine these are internal users indistinguishable from authorized users. An external penetrator who has gained access to someone's account is a masquerader, as is an authorized

user who operates under another user's account.

b. Legitimate Users - sometimes known as misfeasors, these users abuse their authorized access to the computer system and its data. For instance, a user with authorized access to highly classified documents might use a covert channel to convey the contents of the documents to someone who does not have access to them. Many experts feel that the legitimate user is probably the most common computer system intruder or abuser.

c. Clandestine Users - these users are ones who have or can seize supervisory control and can thus operate below the level of auditing or can evade the auditing such as by turning off the audit function.

Donn Parker's and Peter Neumann's SRI Computer Abuse Methods Model describes a more comprehensive classification of eight computer abuses [Neum88a]:

1. External Abuse - passive (to the computer) actions such as eavesdropping, physical waste scavenging, visual spying, espionage, multi-person collusions. These terms are used in the ordinary sense.
2. Hardware Abuse - generally computer-active actions such as equipment theft or damage, tapping of the communication bus, Trojan horse installation, electromagnetic interference. A Trojan horse is "a

computer program with an apparently or actually useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process to the detriment of security." [DoD85]

3. Masquerading - such as impersonation, password attacks, piggybacking, playback and spoofing attacks, telephone network weaving to hide dial-up origin and tail-gating. Password attacks are a user's attempt to login to another's account by guessing at or otherwise deriving (such as by random generation) the victim's password. Piggy-backing occurs when a wire-tapper is able to append his/her own data to a legitimate transmission as it passes by on the wire. Playback and spoofing involve the capturing of a legitimate transmission. Telephone network weaving refers to a user who remotely accesses a computer via a telephone line and proceeds from there to remotely access one or more other systems until his/her point of dial-up origin is obscured and is undiscernable to the destination system. Tail-gating refers to a process's ability to be accidentally or intentionally attached to an incompletely deallocated resource.

4. Preparation for deferred abuse - such as the planting of Trojan horses and viruses. A virus is a special

type of Trojan horse that propagates itself through a system or network of systems [Cohe84].

5. Bypass of intended controls - such as acquisition of unauthorized privileges, unintended reading, writing, or copying, integrity violations, trap-doors, covert channel exploitation. A trap door is "a hidden software or hardware mechanism that permits system protection mechanisms to be circumvented reliably and without detection." The trap door is activated by a special command or key sequence [Gass88], [DoD85].

6. Passive abuse - such as browsing, inference, data aggregation, activity monitoring. Browsing is a random searching through other user's or system directories. Inference refers to a user's deriving, from several facts he/she has gathered, some form of information to which he/she would not normally have access. Data aggregation is accomplished by one or more persons who obtain data at their appropriate classification levels and then combine the data such that the combined data requires a higher classification than the individual parts. Activity monitoring refers to the notion of studying the activity patterns (command usage, program execution, etc.) of a user or group of users in an attempt to derive information about that user or group.

7. Active abuse - misuse of conferred authority, false

or erroneous data entry, denials of service, computer network weaving, and worm attacks. Computer network weaving is similar to telephone network weaving in that the user covers his/her login or access trail so that the computer system does not know the user's point of origin. A worm attack is accomplished by a program which lies dormant in a system until there is sufficient available resources for it to run. A worm program can steal CPU resources and possibly result in a denial of service.

8. Use as an aid to committing a crime or other misdeed - such as using one computer to aid in penetrating another or using a computer to run an illegal drug business.

It is worth noting that not all computer threats are intentional. Certain accidental misuses of the computer system can also have harmful consequences. Such accidental misuse could include system personnel mistakes which can effect all users or user mistakes which result in denial of service to others [Neum88b].

By assimilating information from a wide range of computer crime reports, Allan Clyde [Clyd87] described five basic categories of damage a computer system sponsor may suffer due to computer abuse:

1. Denial of Service - the system becomes inoperable and unusable for some or all users.
2. Information Loss - information managed by the system is destroyed or corrupted.
3. Disinformation - information that is made to be misleading.
4. Information Compromise - information is provided to persons not authorized to receive it.
5. Resource Exploitation - the system is used to promote objectives not authorized by the sponsor.

Each of these threat or abuse classifications [Ande80], [Neum88a], [Clyd88] is an attempt to define the many problem areas faced by computer security. A comparison of these classifications underscores the fact that the abuses are not always easily classified. Neumann notes that threat categories are not necessarily discreet divisions but instead should be viewed in terms of what damage the intruder might cause and how intrusions might be differentiated for detection [Neum88a]. Interestingly, he points out that certain differentiations between threats are probably moot in respect to the fact that the damage is done:

1. External and internal penetrations -- once an external penetrator is in, he becomes an internal threat.

2. Unauthorized and authorized users -- except for recording failed login attempts, the computer does not know the difference.
3. Masquerader and legitimate user -- the masquerader could be an insider or an outsider.
4. Trusted and untrusted users -- these cannot be relied upon as inpenetrable barriers.

1.2 RELEVANT RESEARCH

Intrusion Detection Systems (IDS's) are still an emerging area of computer security, but some serious research in security audit trail analysis has been developed in the past few years. Most Intrusion Detection System implementations are in the experimental stage though there are a few limited analysis tools on the commercial market [Lunt88b], [Clyd87] and at least one in the U.S. Federal Government [Hann88]. This section gives a brief description of some of the major research developments but does not attempt to cover all the valuable experimental supporting research.

J.P. ANDERSON CO.

In 1980, James Anderson concluded a study of security audit trails and the role they play in detecting computer abuses [Ande80]. As mentioned in the previous section, he

classified the different general threats to computer security and for each he offered some possible methods of detecting these threats by analyzing the security audit trail. Of particular significance, he points out that actions by a masquerader constitute an "extra" use of the system. This use would probably be abnormal with respect to the proper user's normal past behavior and could probably be noticed by analysis of audit records.

The Intrusion Detection Model

Anderson's hypothesis, and a great deal of experimental research at Sytek and at the Stanford Research Institute (SRI) International, formed the basis for SRI's Intrusion Detection Model [Denn85], [Denn87]. This model is meant to provide a framework for a general intrusion detection system which is independent, both physically and logically, of the system it is analyzing (the target system). The model is a rule-based pattern matching system. In short, audit records from the target system are matched against statistical profiles of user behavior which are learned by the Intrusion Detection Expert System (IDES). If the current behavior as shown in the audit record exceeds an established threshold of the profile, an anomaly is generated and the system security officer is alerted in real-time.

The model consists of six main components:

1. Subjects: Initiators of activity on a target system - normally users.

2. Objects: Resources managed by the system - files, commands, devices, etc.

3. Audit Records: Generated by the target system in response to actions performed or attempted by subjects on objects - user login, command execution, file access, etc. [See figure 1.1]

4. Profiles: Structures that characterize the behavior of subjects with respect to objects in terms of statistical metrics and models of observed activity. Profiles are automatically generated and initialized from templates.

5. Anomaly Records: Generated when abnormal behavior is detected.

6. Activity Rules: Actions taken when some condition is satisfied, which update profiles, detect abnormal behavior, relate anomalies to suspected intrusions, and produce reports.

< Subject, Action, Object, Exception-Condition, Resource-Usage, Time-Stamp >

Figure 1.1 Audit Record Format.

For efficiency, the audit record format in figure 1.1 is a standard format recognized by the IDES [Denn85]. The tuples are:

* Subject: the initiator of the recorded action.

* Action: a single-object operation the subject performs on the object, e.g., login, logout, read, write.

* Object: the receptor of the action. Types of objects are files, programs, messages, terminals, printers, etc. Subjects can also be objects such as when they are the receptors of electronic mail.

* Exception-Condition: denotes which, if any, exception - conditions [errors] are raised on the return.

* Resource-Usage: list of quantitative elements, where each element gives the amount used of some resource, e.g., number of lines or pages printed, number of records read or written, CPU time or I/O units used, session elapsed time.

* Time-Stamp: unique time/date stamp of the action.

Formally, the sum of the audit records conceptually forms an audit matrix which is very similar to an access matrix [Denn85]. The state of the target system is defined by a triple (S,O,A), where:

1. S is the set of subjects. Subjects perform actions on objects. $S \subseteq O$.

2. O is the set of objects. Objects are acted upon by subjects. Each object is uniquely identifiable.

3. A is an audit matrix, with rows corresponding to subjects and columns to objects. An entry A[S,O] lists the actions that subject S performed on object O. Also listed in A[S,O] is the associated error-conditions and time-stamps of the

actions and the cumulative resource usage of the subject [Denn82], [Denn85].

The IDES prototype discussed in the next section conceptually views the audit matrix from the subject angle as it generates statistical profiles about the normal behavior of the subjects. However, it seems reasonable that one could as easily approach the audit matrix from the object view so that profiles about the normal usage of particular objects could be determined. Stated another way, it should not be significantly more difficult to draw conclusions about all user's behavior with respect to a particular object such as a very sensitive file or program. Denning gives some suggestions on the classes of objects and the types of measures which should be tracked and analyzed [Denn85]. This idea will be explored further in Chapter 3.

The Intrusion Detection Model (IDM) describes the processing of audit records, the production of profiles and the use of the activity rules to determine an intrusion. The Intrusion Detection Model is the framework for SRI's prototype and enhanced prototype Intrusion Detection Expert Systems (IDES's) and almost all other subsequent related research in automated audit trail analysis.

IDES

At the forefront of automated audit trail analysis research, the IDES [Denn87], [Lunt88a], [Lunt88b], [Lunt88c], [Lunt89] is an iterative prototype of a real-time intrusion detection system developed by a team at SRI. The IDES monitors a DEC-2065 running a customized (for security auditing) version of the TOPS-20 operating system. In its next phase, the IDES will monitor several target systems simultaneously. The highly sophisticated IDES is composed of two separate detection entities: Statistical Intrusion Detection and Rule-based Intrusion Detection (the Expert System). It is intended that inappropriate behavior will be detected by one or the other entity, and possibly both.

The Statistical Intrusion Detection component generates the subject profiles for normal subject behavior and compares them with new audit record data looking for significant deviations from the profiles. This strategy is usually effective in detecting masqueraders and authorized users who suddenly and suspiciously depart from their normal behavior, assuming that the authorized user normally maintains a stable pattern of behavior. If, however, the subject's behavior is very erratic or too new to have an accurate, established profile, the Expert System component may be useful. The Expert System contains rules which

characterize intrusions based on knowledge of past intrusion, known system vulnerabilities and the installation-specific security policy.

> The knowledge-base of the expert system contains information about known system vulnerabilities and reported attack scenarios, as well as our [the developer's] intuition about suspicious behavior [Lunt89].

Thus the rules may be specific to the environment but are independent of any particular subject's normal behavior. The Expert System looks for any departures from what it considers normal behavior for any subject. An example of a departure from normal behavior is the set number of failed login attempts by any single user that the monitor allows before it signals the security officer that someone might be conducting a password attack.

The IDES monitors three different types of subjects: users, remote hosts, and target systems. For these subjects, the IDES monitors 36 different measures. See figure 1.2. There are two types of measures - categorical and continuous. The values for each of the measures is used to create profiles and to update them at the end of each time segment (for a subject, this could be a user session).

A categorical measure is a function of some aspect of observed behavior whose range is the set of all combinations of a finite set of categories. An example of a categorical measure is the commands invoked by a user, where the range is all combinations of file names. Another example is the hour of activity by a user, where the range is all combinations of the 24 hours of a day.

A continuous measure is a function of some aspect of observed behavior whose range is the set of real numbers. An example of a continuous measure is the length of a user-session. Another example is the number of lines printed by a user during a session [Lunt88a].

User Measures

1) CPU usage (continuous).
2) Input/output usage (continuous).
3) Connect time (continuous).
4) Audit records generated (continuous).
5) Shift of login (categorical).
6) Location of use (categorical).
7) Location change count (continuous).
8) Command usage (categorical).
9) Command usage (binary) (categorical).
10) Mailer usage (categorical).
11) Editor usage (categorical).
12) Compiler usage (categorical).
13) Directory modification (continuous).
14) Directories accessed (categorical).
15) Directories accessed (binary) (categorical).
16) Errors (continuous).
17) Errors by type (categorical).
18) Hourly use (categorical).
19) Hour of use (binary) (categorical).
20) Day of use (binary) (categorical).
21) Network activity (continuous).
22) Network activity by host (categorical).
23) Network activity by type (categorical).
24) Hourly network activity (categorical).
25) Hourly network activity by host by type (categorical).

Host Measures

1) Host users (categorical).
2) Activity types (categorical).
3) Hourly use (categorical).
4) Hourly use by type (categorical).
5) Bad login attempts (continuous).
6) Hourly bad login attempts (categorical).

System Measures

1) Bad login attempts (continuous).
2) Hourly bad login attempts (categorical).
3) System errors (continuous).
4) System errors by type (categorical).
5) Hourly system errors (categorical).

Figure 1.2 The measures used in the IDES. [Lunt88a]

A subject profile consists of four components:

1) Effective Count Vector (Effn) - this holds the count of the number of time segments (i.e. user sessions) for each measure for that subject.

2) Mean Vector (Mean (x)) - this contains the historical mean value for each measure, x, for the subject.

3) Covariance Matrix - the values in this matrix inter-relate all the measures observed for that subject. The elements are determined by:

$$Cov(x,y) = 1/n(\sum_{i=1}^{n} v_i(x)\ v_i(y)) - Mean(x)\ Mean(y).$$

where x and y are measures, $v_i(x)$ and $v_i(y)$ are values for the ith time segment of the measures under consideration and n = min(Effn(x), Effn(y)).

4) Inverse of the Covariance Matrix.

Anomaly detection is performed for all subjects as each audit record arrives. The statistical procedure of the IDES evaluates whether a particular observed measure value deviates relative to the observed values for all the other measures, not just with respect to that measure considered independently. To accomplish this, the IDES uses a Composite Test:

$$t2 = (X - \overline{X})\ C^{-1} (X-\overline{X})^T$$

where X is the continuous measure vector for a user's session, $\overline{X}$ is the mean vector from the user's profile and C^{-1}

is the inverse covariance matrix also from the user's profile. An anomaly exists if the value of t2 falls outside the 95% probability range of its distribution. Anomalies are reported through a sophisticated security officer interface. Using the interface the security officer has many options such as monitoring activity at the user-session level, making direct queries to the IDES database, and customizing the various parameters used to control how the IDES monitors subjects (such as the number of failed login attempts allowed).

Audit

The Audit system was developed by Clyde Digital Systems [Clyd87] who classify it as an Insider Threat Identification System founded on (1) internal system surveillance. The other four basic components are: (2) analysis of the surveillance data by an expert system, (3) identification of perpetrators using the expert system, (4) tools for detailed damage assessments and (5) support capability for recovery. The Audit system monitors VAX/VMS machines. The surveillance system captures all user interaction with the system and uses 14 risk factor tests to identify high-risk users and record their activity.

MIDAS

The Multics Intrusion Detection and Alerting System (MIDAS) is a real-time rule-based Intrusion Detection System developed at the National Computer Security Center (NCSC). MIDAS monitors activity on the NCSC's networked mainframe, DOCKMASTER, a Honeywell DPS-8/70 Multics System [Hann88]. Its auditing components exist on the target machine while its knowledge bases , statistical database and system security officer interface are installed on a separate Symbolics Lisp machine. The MIDAS rulebase contains three types of heuristics used to analyze the audit data for intrusions:

* Immediate Attack - these represent a priori rules about what constitutes an intrusion.
* User Anomaly - these make use of statistical profiles of user behavior.
* System State - these also use statistical profiles to characterize behavior of the entire system.

The MIDAS system is currently in operation at the NCSC and continues to be improved.

TACAUD

The Network Auditing Usage Reporting System (NAURS) operates with the Terminal Access Controller (TAC) Access

Control System for the ARPANET and MILNET [Lunt88b], [Neum89]. The ARPANET was created by ARPA, now DARPA, the (Defense) Advanced Research Projects Agency of the U.S. Department of Defense. MILNET, a military network, was established later using the ARPANET technology. The TAC auditor (TACAUD) runs on the NAURS system and monitors network usage initiated from TAC's such as logins, logouts, connects and disconnects to the hosts. Note that TACAUD monitors terminal connections to a host, but does not monitor subsequent TELNET (remote) connections from that host to other hosts. Therefore, the monitoring system can effectively lose track of a user's activity on the network. The nature of this problem is discussed further in Chapter 2. Individual hosts monitor their own usage. The system is currently rule-based though statistical profiling may be added in the future.

Network Security Monitor

The Network Security Monitor (NSM) is a product of on-going research at the University of California at Davis and Lawrence Livermore National Laboratory [Mans88]. Goals of this research are to identify vulnerabilities created by connecting computers into networks and how exploitation of these vulnerabilities might best be identified. The NSM is concerned with the misuse of the information packets which

are flowing around the network. Some of the identified threats are: rerouting of packets; data modification within the packets; packet delay; flooding/jamming; imitation by altering the source address of the packet. The NSM places packet catchers on each separate network line to monitor the traffic. The current model runs on a Sun workstation and monitors an Ethernet LAN. Currently under consideration is the question of what granularity levels of monitoring are most effective for detecting intrusions.

While this relatively new field of intrusion detection has seen a flurry of activity in developing audit analysis tools for both government and commercial use, there are still many open questions and needs for future research. Of the several examples given in this section, almost all rely heavily upon Anderson's hypothesis about abnormal behavior and on Denning's Intrusion Detection Model. For simplicity, we will refer to all systems of this type as Intrusion Detection Systems (IDS's). The current IDS's all monitor single systems. An important open question is that of how to apply the Intrusion Detection Model to a network of interconnected computer systems. This research provides a model of network intrusion detection and a framework for its continued study.

1.3 RESEARCH GOALS

In approaching the problems associated with monitoring events on a network of interconnected computer systems, one must first understand how a network differs from a single system and how these differences can benefit an intruder. One major difference is that a user on one machine has the potential to access information, execute programs or otherwise manipulate system components on all the other machines of the network. This issue and others will be discussed in Chapter 2, The Problem. Following in Chapter 3, The Network Monitor Model, is a description of our approach to handling those issues. Included is the description of a two level Network Monitor Model which monitors at the LAN and at the internet levels. This chapter also discusses a problem of the statistical model suggested by Denning's Intrusion Detection Model and currently used in some intrusion detection implementations and suggests a different statistical approach. Chapter 4, Results and Conclusions, presents the significant findings of this research and suggests direction for future study in this area.

As mentioned earlier, because no published research has yet undertaken the complex task of monitoring a network for intrusion detection, there were no pre-conceived ideas on

how this should best be done. Thus, the goal of this research was to develop a valid, generalized approach and framework for continued study in network intrusion detection. Included in this framework is a proposed two-level Network Monitor Model and several specific issues identified for further study. The Network Monitor Model is intended to be general enough that it can be applied to any size and type of network, possibly even extending the two-level model into three or more levels as dictated by the network topography.

CHAPTER 2

THE PROBLEM: INTRUSION DETECTION ON NETWORKS

2.0 INTRODUCTION

Current knowledge of how to automatically analyze audit data for a single site system provides some degree of protection; however, it is not enough to consider a single system in isolation. In general, the organizations with the most need for information security are those with a great need for interaction among their sub-units and with external sources as well. As intercommunication among computer systems can only be expected to increase in the future, ways must be found to detect insecurities on connected systems.

When two or more computer systems are connected by a cable, switched telephone line or satellite, in theory a user on one system gains access to all of the computing power and resources in the network. Security policies need to be adopted to control interaction between connected computer systems; auditing can help to ensure enforcement of these policies. However, the concept of auditing and the analysis of the audit data collected becomes increasingly

more complex proportional to the size and complexity of the monitored network. In this research, we apply Denning's Intrusion Detection Model [Denn87] to all types of networks, whether they are local area networks (LAN's), sets of connected LAN's known as wide area networks (WAN's) or some other variation of connected systems. This problem is not merely a trivial expansion or multiplication of the current monitoring tools such as those described in [Lunt88b], [Hann88] and [Clyd87]. Intercommunication between connected systems adds another, more complex, dimension to the intrusion detection model. The purpose of this chapter is to address some of the more significant issues encountered in applying intrusion detection methods to a network environment.

Section 2.1 presents a basic overview of networks and their terminology. Section 2.2 discusses some of the problems of monitoring a network with Intrusion Detection Systems. Section 2.3 describes the problems of monitoring a system having remote file access, remote file transfer and remote login facilities. Section 2.4 discusses the problem of collusion in regard to network intrusion detection. Finally, section 2.5 presents our approach to the problem of monitoring networks.

2.1 NETWORKS

A computer is defined as any device capable of storing and processing information and of communicating with other computers if linked by a network [Walk85]. A computer network is defined as an interconnected collection of autonomous computers [Tann88]. Interconnected here means that there exists no master/slave relation between any two computers of the network. We will often refer to communicating computers on a network as hosts.

A computer network differs from a distributed system in that the user must normally deal explicitly with different hosts on the network whereas in a distributed system, the existence of multiple processors is transparent to the user [Tann88].

The hosts of a network are connected by a communication subnet which can be one of two basic types [Tann88]:

1) Point-to-Point channels. In this case, messages (packets) are passed in their entirety from one intermediate computer (switch) to another until they reach their destination. This is also known as packet-switched. Almost all wide area networks are of this type.

2) Broadcast channels. In this case, there is just one communication channel shared by all the

hosts of the network. Packets broadcast are received by all hosts but are discarded by those that are not the intended recipients. Most local area networks are of this type.

Network Architecture

Most networks are organized into a hierarchy of layers each of which provides a service to the layer above it. The provision of services between any two layers is proscribed by a set of rules and conventions collectively known as a protocol. A commonly used layered network model is the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model shown in figure 2.1.

Cryptography, an important part of computer security, is often handled by the presentation layer. The auditing which we will refer to in this research will likely be accomplished at the network layer.

Types of Networks

There are significant differences between local area networks (LAN's) and wide area networks (WAN's) or other types of internets with regard to developing an intrusion detection system. Local area networks consist of connected systems (hosts) and other peripheral devices physically

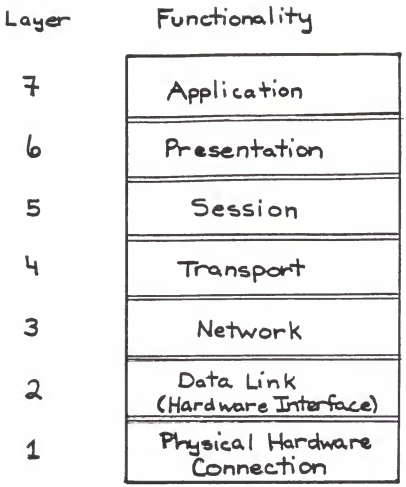

Figure 2.1 The ISO OSI Reference Model.

located within a relatively small area ,e.g. a building. Each of the components of a LAN can be subject to the control of a central authority which establishes and enforces the security policies of the LAN. A department of a large company or university may be allocated a LAN. Several of these LANs may be connected at the Data Link layer by a relay called a bridge or a gateway. This configuration is known as an internet. The term internet describes any collection of two or more packet-switched networks interconnected by gateways plus the protocols which enable the networks to function logically as one large network. The Internet is an internet operated by DARPA which uses the TCP/IP protocols.[Come88]

A WAN is a type of internet that is characterized by great physical distances between its components. These components are normally LANs but may be individual hosts. The components of a WAN are connected by gateways which are specialized computers that operate at the Network Layer to enable messages to pass between heterogeneous LANs and hosts. A WAN is often a more loosely coupled network whose member LANs may be dissimilar and unrelated. Further, there may be no central authority over the WAN other than for network administration purposes and consequently no common security policies or means of enforcing those policies.

It is projected that the Intrusion Detection Expert System (IDES) being developed at SRI is capable of monitoring audit data from more than one target system at a time.[Lunt88a] In the IDES model, each system performs its own audit functions and then passes standard format audit records to the physically separate and independent IDES. This may be feasible if the IDES treats each system's data separately, as if each system logically had its own IDES. Both for physical and logical reasons, one can not expect one intrusion detection system to monitor all of the systems on a network of interconnected LANs. If we apply the IDES concept to a WAN we can visualize an internet of LANs in which each LAN has a dedicated Intrusion Detection System (IDS). Each LAN is then responsible for enforcing and monitoring its own security policies irrespective of any other system with which it communicates. Likewise, any individual host which communicates directly with a WAN conceptually would have its own IDS. In this model the IDS's do not communicate with one another. A proposed monitored internet is shown in figure 2.2.

2.2 MULTIPLE IDS's ON A SINGLE NETWORK

If a single intrusion detection system is to be able to effectively process the audit data generated by several target machines, there are at least two important unresolved

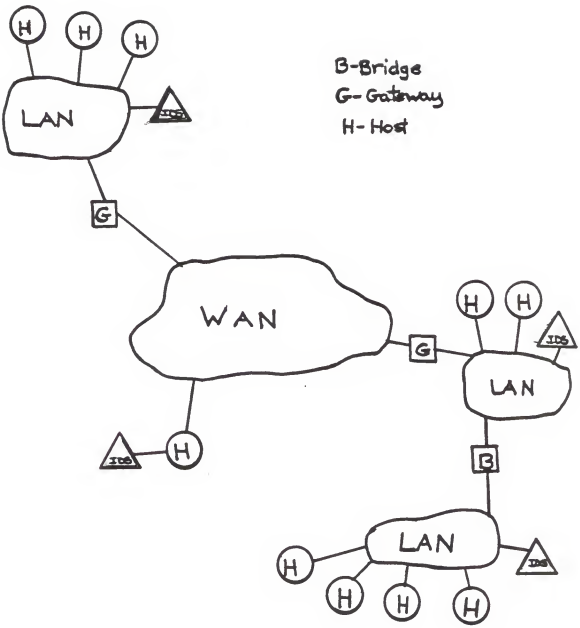

Figure 2.2 A Proposed IDS-Monitored Internet Configuration.

issues to consider:

(1) How many target systems should one IDS monitor?

(2) How to handle multiple instances of user profiles?

In regard to the first issue, for the purposes of this paper, we will assume that each LAN is monitored by a single Intrusion Detection System.

Multiple Instances of User Profiles

Because each LAN is monitored by a single IDS, we will further assume that each IDS is completely independent of the others. Each IDS maintains its own databases of statistical profiles, short-term user data, and expert system rules and knowledge. This is critical to the security of the IDS databases whose contents must be protected from unauthorized alteration and from unauthorized disclosure of personal information. Therefore, as opposed to considering any type of distributed monitoring, each IDS would maintain its own databases of statistical profiles, short-term user data, and expert system rules and knowledge.

Independent IDSs lead to a problem -- the existance of multiple profiles for the same subject (or object). Suppose, for example, that a subject owns accounts on more than one system and that at least one of those systems is monitored by a different IDS than the others. Each of these

IDSs would have a statistical profile of that user representing his/her actions with respect to the target systems that the IDS is monitoring. Each IDS would have no knowledge of the subject's profiles which exist on the other IDS's.

A subject may use the existance of these profiles, and the fact that the IDSs do not communicate among themselves, in an attempt to confuse the audit analysis. For instance, if a subject works within the limits of his/her various individual profiles on different machines, no reportable anomalies of any consequence are detected. But subjects may be able to commit abuse over the collective work done on all the systems. Two areas in which subjects could create this type of confusion are:

(1) remote access

(2) collusion

These areas will be covered in the following sections.

2.3 REMOTE ACCESS

Processes can communicate between machines on a network by using remote access calls and file transfer protocols. These forms of communication have the potential for a subject to abuse the systems. A subject is able to be logged into one system and by using a remote system call may

execute commands and perform functions such as reading and copying files on a remote system without directly logging into it. This description is characteristic of the UNIX operating system although we assume most other operating systems provide some other similar form of remote system calls. Throughout the remainder of this paper, all references to operating system features will be based on the UNIX system because this is the one with which we are most familiar. Most of the information regarding UNIX commands was taken from [UNIX86]. A problem is that the IDS monitoring the remote system is unaware of the original source or of the subject profile in the original source environment. Three classes of remote calls which are susceptible to this problem are:

(1) remote login

(2) remote shell

(3) file transfer

Remote Login

With a remote login service, a user is able to log onto another system from the system which he/she is already logged onto if the user is authorized to use the remote system. In the UNIX systems there are two programs to accomplish remote login: *rlogin* and *telnet*. An example of each command usage is given in figure 2.3. Using the rlogin

command, the user must explicitly execute the login process on the remote machine. Once successfully logged in, the user's shell is invoked, the user is allocated a pseudo-terminal and can function as though the login was local. Shell is a term refering to a command interpreter process which is created at login for each user. A pseudo-terminal describes the operating system entry point that allows a running program like the TELNET and rlogin servers to simulate a terminal [Come88]. Although the user's input is actually originating at the source host, it is being propagated through the pseudo-terminal to the remote host for execution. The user can continue to issue remote login commands from a remote login allowing the situation illustrated in figure 2.4.

(a) % rlogin machine2

(b) % telnet machine2

Figure 2.3 Examples of the UNIX rlogin and telnet commands.

Figure 2.4 The Result of Successive Remote Logins.

From a terminal at host A, the user can list files on system D for example. The command originates at A, emanates thru B and C and finally executes at D, while the output follows the same path in reverse. Events are audited at each system based on what that system knows about the transaction, the user and the system status. In the UNIX environment, as well as several other operating systems, each host only knows about the host that logged into it, not any host previous to that login. Host C knows that B issued a remote login, but does not know that the user is actually originating from A. One can imagine that several remote logins over a large WAN, for example, would effectively cover the trail of a possible intruder making detection and apprehension very difficult. This problem would be greatly compounded if some of the logins were authorized and some were masquerades.

Referring to figure 2.4, consider Intrusion Detection Systems that are monitoring the systems. Each contains a profile of the user that pertains only to that user's behavior on that system. Conceivably, from a terminal at A, the user could cause an event to occur on D with no ill effects while that same event if it occurred on A would have caused an anomaly. This is because the event described behavior that was in keeping with the user's (or the group's or system's) profile on D. The user could use the

knowledge that A and D (and B and C) profile him differently and thus each expects different behavior. A simplistic example of this problem can be depicted as follows:

1) User T's profile of his behavior on machine K "allows" him to print 1500 lines per session before the threshold is exceeded and an anomaly is reported by the IDES.

2) User T's profile of his behavior on machine H "allows" him to print 3500 lines per session before indicating an anomaly.

3) Machine K and machine H are on separate, but connected LAN's which are monitored by different IDS's.

4) User T wishes to print out a file containing 3350 lines without raising an anomaly or attracting suspicion. Therefore, user T logs into machine K, remotely logs into machine H and prints the file. The IDES at machine H does not note any unusual behavior regarding the printing. Machine K merely notes a remote login to H but is not aware of the printing. Thus user T has performed a function at K which normally would have resulted in an anomaly but which goes unnoticed in this case.

A further complication of this scenario would result if the user at A chose to print information contained in D on printers located at B and C. Suppose that the user at A wished to perform an action like this which would not fall

within the thresholds of the user's profile or that of the user's group at A. If the user knows that the action would fit the individual or group profile at D, the user could perform the action at D from the remote system A. The user could also provide this information to others at B and C with functions such as remote file transfer or remote shell.

Remote Shell

Rsh (remote shell), a variation of the rlogin program, is very similar in that it allows the user to execute a command on a remote machine or system. However, unlike remote login, the user never explicitly logs onto the remote system. The user's shell on the remote machine is invoked and therefore all the user's actions on the remote machine can be audited and properly attributed to him/her. The user can execute a single command sequence which upon execution completion at the remote host returns the user to the originating shell. It is important to note that control is not returned to the local shell until the remote command has terminated. This point may become important when using time-event ordering to determine a remote user's point of origin as we suggest in Chapter 4. Because rsh does not prompt for a password, it can be used in programs as well as from the keyboard. Figure 2.5 gives an example of the rsh command usage.

```
% rsh machine2 ls
```

Figure 2.5 This example of rsh would list (ls) all the files in the user's home directory which resides on a different machine than the one the user is logged into.

Here again, the user can issue successive remote shell commands so that he/she is effectively working on a machine several systems removed and thereby greatly confuse the system as to his/her true disposition. See the example in figure 2.6.

```
% rsh machine2 rlogin machine1
```

Figure 2.6 This command would log the user back onto the machine from which he/she is issuing the command.

File Transfer

File Transfer programs allow the user to copy files from one host machine to another on the network. The UNIX operating systems contains several different forms of this type command: *rcp*, *uucp* and *FTP*. The rcp (remote copy) command requires that the local user name must exist on the remote host and allow remote command execution via *rsh* (remote shell). This command is capable of handling third party copies where neither the source nor target files are

on the current machine [UNIX86]. See the example in figure 2.7. FTP (File Transfer Protocol) is an Internet protocol that allows authorized users to log into a remote system, list remote directories, copy files to or from the remote machine, and execute a few simple commands remotely. FTP is more complex than TELNET in that it uses the TELNET protocol for its control connection, it allows a user to access multiple machines in a single FTP session and it maintains separate TCP (Transmission Control Protocol) connections for control and data transfer. FTP was designed to be used by programs but can also be used directly by users. Like rcp, FTP can also handle third party file transfers [Come88]. UUCP (Unix to Unix Copy Program) allows one UNIX system to copy files to or from another UNIX system over a single (usually dial-up) line.

```
% rcp machine2:file2 machine3:file3
```

Figure 2.7 An example of copying a file between two machines from a third machine (machine1) using rcp.

To further complicate the situation, any of the three types of remote access commands (remote login, remote shell and file transfer) can be used in combination to cause considerable confusion for any audit trail monitor. The key problem involving these remote access capabilities is that there is a great potential for the Intrusion Detection

Systems to become confused about the actual location, time or even the object of a subject's action thus resulting in an incorrect assessment of the normality of the behavior.

2.4 COLLUSION

Collusion is a secret agreement or cooperation for a fraudulent or deceitful purpose [Merr74]. Specifically we refer to collusion of efforts between two or more subjects who are attempting to gain unauthorized access to information by pooling their accessing abilities. For example, if two subjects are each authorized access to two different portions of a statistical database, they might be able to combine the information they each can legally extract to form a tracker with which they can illegally obtain details about specific entries in the database. The problem of collusion, though, is not limited to databases. Several subjects could agree to use their systems as a ruse in order to confuse the monitors and draw attention away from an actual intrusion. Or, they could all cooperate to flood the network with auditable events and effectively cause a denial of service condition.

Closely related to collusion between subjects is the problem of information aggregation. One or more users can obtain data which is classified at a level appropriate to

their clearances and combine this information such that the compilation of knowledge should actually require a higher security classification or protection than do the individual data elements [Vaug88].

Collusion Classification

While there are infinitely many possible varieties of collusion or information aggregation between intruders, we recognize four possible scenarios which would determine classifications of collusion attempts as they are monitored by an IDS:

(1) All intruders working on the same host.

(2) All intruders working at the same site (LAN) but on different hosts.

(3) One intruder who works from multiple hosts on the same or different LANs (information aggregation).

(4) Two or more intruders working from multiple hosts on different LANs.

The problem here is that the would be intruders are taking advantage of the fact that their dispersed activity is being monitored by different intrusion detection systems. Individually, their activities may not violate the thresholds of their user profiles and no anomalies will be reported. However, together these "colluders" are able to aggregate their information to gain knowledge which would have caused an anomaly if sought by an individual. The

results of this concept are very similar to the remote access problem, although we may be more successful in detecting and deterring intrusion by remote access than in detecting collusion. The one scenario in which we feel we have the best chance of detecting collusion is that of (3), one subject working from multiple hosts on the same or different LANs. Of course, if this scenario were clouded by numerous remote access calls, detection could become very complex.

2.5 THE PROBLEM APPROACH

The architecture and design approach presented in the following chapter considers the problems of monitoring in the network environment. Using the Access Matrix Model, the problem is approached at two levels, the local (LAN) level and the global (WAN) level. The approach is also from a somewhat different perspective than that of previously published research [Lunt88b], [Hann88], [Clyd87].

When a system in our model collects audit data, it is in effect forming a three-dimensional table which has a structure very similar to the Access Matrix Model [Denn82], [Denn85]. We will call this table an Audit Data Matrix (ADM). The indices of the ADM are subjects, objects and time. Entries into the table are events, the operations

performed by the subjects on the objects.

The first level of our Network Monitor Model, the Local Area Network Intrusion Detection System (LAN-IDS), will draw on data from the subject view of the ADM. Using this data, it will construct subject profiles for all the users of the systems in that LAN. This is congruent to the approach of existing IDS prototypes as discussed in Chapter 1. The LAN-IDS should be capable of detecting most of the intrusions on the LAN which it monitors when those intrusions originate from within that LAN. More complex intrusions such as those involving remote accesses from outside the LAN will probably be detected with the aid of the second level of our model.

The second level of our Network Monitor Model, the Global Object Monitor, will draw on data from both the subject and object views of the ADM which will be supplied by the target systems both on the LANs and as individual hosts. The Global Object Monitor will construct combined object profiles for selected sensitive objects of the internet as well as combined subject profiles. The object profiles will be based upon the same concept as the subject profiles but will be derived from somewhat different measures. The Global Object Monitor will have a 'total picture' of the network. By combining the total picture with the object profiling, the Global Object Monitor will

have the ability to detect some of the intrusions which would not be evident at the LAN-IDS level. A conceptualization of this approach is given in figure 2.8.

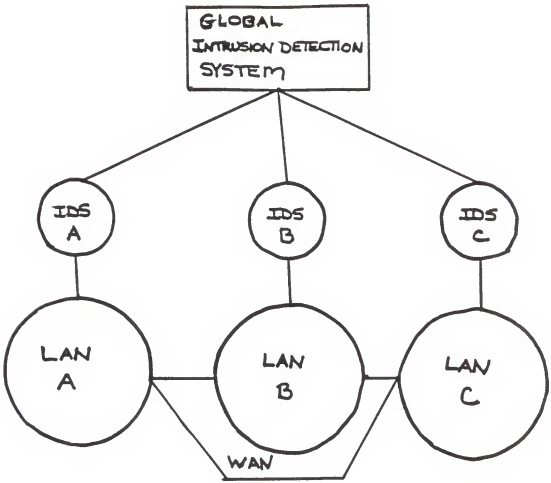

Figure 2.8 The Network Intrusion Detection Monitor Concept.

CHAPTER 3

A NETWORK MONITOR MODEL

3.0 INTRODUCTION

The previous chapter discussed some of the problems of monitoring activity on networks. In this chapter, we present a model for monitoring networks which consist of interconnected Local Area Networks (LAN's).

An underlying premise of the model is that LAN's and large internets of connected LAN's behave differently and therefore will have different monitor requirements. Our Network Intrusion Detection Model provides for monitors at two levels of the network. The Local Area Network Intrusion Detection System (LAN-IDS) will monitor the activity of a single target system or two or more systems connected as a LAN. The second level of intrusion detection will be handled by the Global Object Monitor (GOM) which will oversee the collective activity of an internet or Wide Area Network (WAN). The Network Intrusion Detection System Model can be conceptualized as shown in figure 3.1.

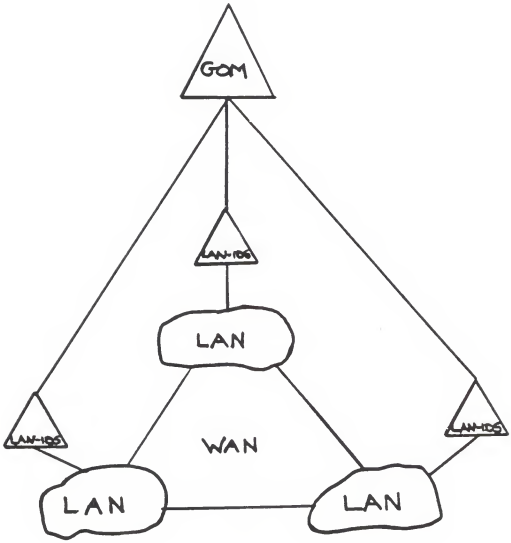

Figure 3.1. The Network Monitor Model.

The two levels of monitors differ in their objectives, their composition and their capabilities. Each of these differences will be discussed in this chapter. While the LAN-IDS can be considered a complete self-contained monitor for a LAN, the GOM is sustained by input from the LAN-IDS's and complements their capabilities in order to provide a more secure network.

A general description of the network monitor model is given in section 3.1. Section 3.2 contains some comments on the statistical model used in Denning's Intrusion Detection Model. Some suggested object measures which should be monitored are given in section 3.3. Sections 3.4 and 3.5 provide descriptions of the two different levels of monitors. We make the following assumptions concerning the networks:

ASSUMPTION 3.1: Each operating system running on the network has an adequate audit capability which records all security relevant events (see Definition 3.0) as dictated by the network controller (see Definition 3.1).

ASSUMPTION 3.2: On each system, it will be possible to determine the source system of a remote access.

ASSUMPTION 3.3: Each operating system running on the network has adequate cryptography capabilities to encrypt all communication from them to their LAN-IDS's.

3.1 A GENERAL DESCRIPTION OF THE MODEL

A conceptualization of the two-level network monitor model was given in figure 3.1. The two levels of the model are meant to work together to detect any breaches of the security policies of each of the LAN's as well as the security policy of the interconnected network. The greatest responsibility for intrusion detection lies with the LAN-IDS while the GOM attempts to detect those intrusions which elude the capabilities of the LAN-IDS's.

Each LAN or separate host on the WAN is monitored by a LAN-IDS. The LAN-IDS monitors all the systems on the LAN in much the same way that the SRI IDES prototype monitors a single system. A key difference is that in addition to monitoring subject (i.e. user) behavior, the LAN-IDS amalgamates the audit data from all the separate systems and detects intrusive behavior which transgresses system boundaries. It does this by forming composite profiles for all subjects and for certain exceptional objects, such as those objects with existing copies on more than one system of the network. The need to monitor objects as well as subjects was first mentioned in [Ande80] and was later discussed in greater detail in [Denn85]. Because the set of objects within a system may vary often and may grow to great proportions, it would be impractical to try to profile every

object. This is different from the set of subjects, such as users, who's size is controlled by the system administrator. The choices of the exceptional objects and the relevant events associated with those objects are dictated by the network security policy of the network controller. Some of the possible object choices would be highly sensitive files, critical databases or employee payroll records.

> DEFINITION 3.1: RELEVENT EVENT. A relevant event for auditing is that subject activity upon an object which corresponds to a measure used in the object's profile. An example is a read of an object if that object were being profiled and one of the contributing measures was the total number of reads of that object.

> DEFINITION 3.2: NETWORK CONTROLLER. For this research we consider a LAN or WAN network controller to be the network owner or body of authority over all aspects of network usage to include security policy, communication protocol, network configuration, etc.

Besides detecting unusual subject behavior, as is now done by the IDES, the LAN-IDS notes an anomaly when it detects any significant deviation from the normal use of an object. Use in this context refers to any of the actions of which an object is the recipient, i.e. read, write, execute, etc. For example, if a highly sensitive database is suddenly accessed 50% more often in a day than usual, this would obviously constitute an abnormal use of that database.

The GOM provides the additional support necessary at the WAN level by merging subject and object profile data from all the interconnnected LAN's and individual hosts. There is only one GOM for a WAN. It will necessarily process very large volumes of data and therefore the model does not attempt to detect intrusive behavior in real-time, unlike the LAN-IDS. This is acceptable in view of the different detection objectives of the GOM. The GOM receives all its input from the LAN-IDS in batches at regular intervals, possibly once per day during non-peak hours. The GOM will construct composite profiles of all subjects and of the most critical objects present in the network and will attempt to determine abusive behavior which results from such tactics as collusion, data aggregation and computer weaving as discussed in Chapter 2.

3.2 A COMMENT ON STATISTICAL ANOMALY DETECTION

As described in Chapter 1, Denning's Intrusion Detection Model and the IDES employ a variation of the mean and standard deviation statistical model and Chebyshev's inequality. This variation places greater weights on more recent values (by applying a decay factor to the data) and considers the correlations among the different measures. This method has detected anomalous behavior with acceptable false positive rates as reported by SRI [Lunt88a]. However,

this statistical method may not be the best choice [McNu89]. If the measured behavior values, such as a user's behavior, is always distributed normally about the mean and each behavior value is independent of any of the previous values then the mean and standard deviation model could be an appropriate method to use in detecting unusual behavior. However, it seems very unlikely that a user's behavior pattern would be normally distributed and it seems much more likely that each user action would indeed be influenced by other actions which had previously occured. Therefore, any statistical test for deviant behavior should not look at the current behavior as an isolated situation but instead should consider the current behavior in relation to recent past behavior by the same user.

Denning's choice of methods seems to be based on the assumption that the behaviors as recorded in the audit records are discrete events independent of the previous events and the order and time of their occurance. This method is depicted in figure 3.2.

Even if we do assume that a user's behavior represents a normal distribution, there are two problems with this model as it is used for anomaly detection:

1) The possibility of an excessive false positive rate, i.e. finding acceptable behavior to be anomalous.

This equates to a Type I error if we take the Null Hypothesis to be the user's historical mean behavior.

2) The greater possibility of producing false negatives. That is, failing to determine that an activity is anomalous, a Type II error.

The chance of producing false positives increases as the distribution of a user's activity moves away from the bell shape [McNu89].

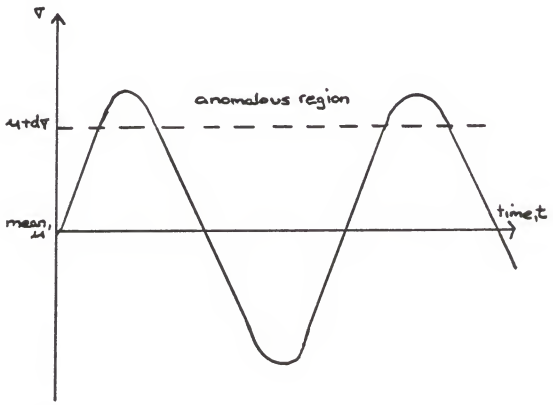

Figure 3.2. User behavior using the mean and standard deviation model. Activity falling above the dashed line is considered anomalous. The curve represents the user's profile; the dashed lines represent the region within which the standard deviation of the behavior over time should fall with a probability determined by Chebyshev's inequality.

A better solution to these problems might be a time-series statistical model. A Time-Series is the "data available for the development of a forecast" ... "in the form of a sequence of dated observations." [Vatt78] These observations are made at regular intervals of time on the variable to be forecast. A time series of past values can be decomposed into component factors (i.e. the relative effect of cyclical factors, the effect of seasonal factors, the effect of unexplained variations, etc.). These components are then extrapolated into the future to form the basis (through recombination) for a probabilistic forecast [Vatt78]. A user's past behavior would be used to forecast his/her behavior that we could expect to see in the future. Applying a time-series model to characterize abnormal user behavior might look as in figure 3.3. Note that the dashed lines which represent the boundaries of the acceptable behavior range now vary directly with the user behavior distribution rather than remaining static as depicted in figure 3.2. Furthermore, a time-series model may be expected to be more sensitive to minor behavior changes than the mean and standard deviation model.

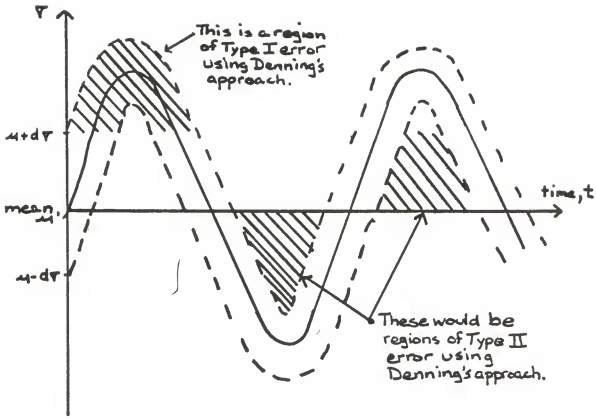

Figure 3.3. A Time-Series Model of User Behavior.

A decay factor, $\gamma = 0.9862$, is used in the IDES enhanced prototype to give the profile data a half-life of 50 days and thereby put greater emphasis on the user's most recent behavior [Lunt88a]. This aging seems to be an attempt to correct for the problem of user behavior changes over time. However, this method is probably not as effective as a time-series model for tracking user behavior trends. In addition, this decay factor benefits the user who plans a future attack and who therefore gradually shifts his/her behavior and profile toward the planned deviant behavior. Instead, a time-series model would probably detect even those gradual and slight variations in behavior and the attack might be thwarted.

The time-series model applied to statistical anomaly detection can be expected to determine the true status of the system security with greater accuracy than the mean and standard deviation model. In [Denn85] and [Denn87], Denning rejects the time-series model as being more costly [perhaps in terms of processing speed and memory use] than the mean and standard deviation model. However, this cost may not be significant compared to the importance of detecting actual intrusions and protecting individuals against the fear of false accusations.

3.3 OBJECTS AND MEASURES

In Chapter 1, a list of the 36 subject profile measures currently implemented on the SRI IDES is provided (see figure 1.2). Each of these measures would also be valid for the LAN-IDS. Additional measures need to be included in order to adequately track object usage and to account for inter-network communications. Chapter 1 also outlined the classes of object profiles suggested by Denning in [Denn85]. These classes are:

1) Command or Program Execution Profiles
2) File-Access Profiles
3) Database-Access Profiles
4) Other types such as system dependent or user-defined object types.

Denning also suggested measures and objects which should be monitored in each of these classes.

Selection of the object types to profile and the measures to monitor for the Network Monitor Model would depend upon the security policy of the monitored network and on an evaluation of current security risks for that network. Further, because of the different purposes of the LAN-IDS and the GOM, these two levels would have different sets of profiled objects and measures. Specifically, the profiled objects and measures of the GOM would be a subset of those for the LAN-IDS's.

At the LAN-IDS level, the objects which should probably be profiled include all those which are highly sensitive or otherwise security relevant, those that are reserved for privileged users (i.e. superusers) and those whose usage tends to reflect the general status of the system [Denn85]. Examples of these for the LAN-IDS include:

* Login and logout programs.
* Change password and access programs.
* Editors, compilers, linkers, mail programs, document formatters, and utilities.
* Password files.
* All files with authorization data.
* Audit programs.
* Remote login and file transfer programs.

Those objects which should be profiled at the GOM level are those that are:

1) highly sensitive or security relevant and are remotely accessible from one or more different systems within the same LAN or on a different LAN; or are
2) existing as copies on more than one system of the network. An important example of this would be file-access profiling of databases.

Most of the measures suggested by Denning would be applicable to these objects at both the LAN-IDS and GOM levels. These measures include:

* Execution frequency.

* Resource usage (CPU,I/O) per execution of a particular program.

* Execution denials.

* Read, write, create and delete frequencies for a particular file.

* Numbers of failures of those reads, writes, creates and deletes. [Denn85]

Following are some measures which should be added to an intrusion detection system which monitors a network:

* Remote execution frequency - a continuous measure of the number of executions of a program from remote systems.

* Remote access frequency - a continuous measure of the number of accesses (read, write, create, delete) to a file from remote systems.

* Location of command issuance - a categorical measure of the number of times per time period that a program is executed or a file is accessed from a particular system (or location). The range is all the systems connected to the network.

* Remote login frequency - a continuous measure of the number of remote logins to each network host.

* Remote login frequency - a categorical measure whose range is all the hosts on the network. It measures the number of times a subject on one host remotely logs onto the profiled system.

We suggest that the time period for the measures of an object be one hour. This period length would depend on the activity of the network, especially in respect to the amount of activity that is associated with the monitored objects. A LAN with especially low activity might chose a

longer time period such as one day, but the normally consistently high activity of a WAN dictates the shorter time period for profiling at the GOM level.

The values for the measures are derived from the audit matrix as discussed in [Denn85], [Denn87] and Chapter 2. Though actually an implementation detail, a modification of the audit matrix concept might better serve the purposes of the Network Monitor Model. The modified audit matrix would be a three-dimensional matrix with *time* added as the third dimension. This is shown in figure 3.4. The three-dimensional audit matrix concept allows one to more easily visualize how to extract the data about a particular object over each time period (one hour).

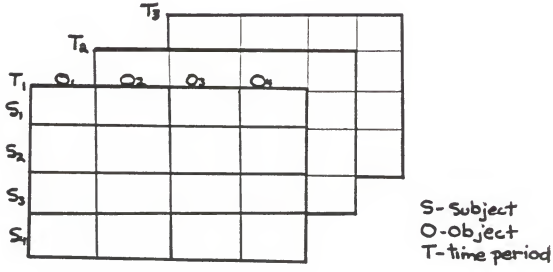

Figure 3.4. The Audit Data Matrix in Three Dimensions.

A simplistic demonstration of the derivation of an individual object profile at the LAN-IDS level is given below. This example follows the methodology used in the IDES [Lunt88a]. The purpose of this example is: (1) to show that an object profile can be produced in much the same way and from the same audit matrix data as the subject profiles; and (2) to illustrate the methodology used in the IDES.

Object O is a database relation which can be read or written. An error is noted if it occurs during one of the reads or writes. The following table lists these continuous measure values for the 24 1-hour time periods of Day 1.

Hour	#Reads	#Writes	#Errors
0	1	1	1
1	2	0	0
2	2	0	0
3	1	1	0
4	0	0	0
5	0	0	0
6	0	1	0
7	2	5	1
8	12	20	2
9	10	4	3
10	18	16	5
11	7	8	1
12	2	2	0
13	21	10	2
14	28	25	4
15	26	21	7
16	12	12	1
17	10	9	1
18	5	7	2
19	6	3	1
20	11	10	3
21	13	15	3
22	7	3	0
23	2	2	0

Table 3.1 Reads, Writes and Errors on Object O on Day 1.

(1) The Count is the number of time periods that a measure was audited.

	#Reads	#Writes	#Errors
Count =	24	24	24

(2) The Sum is the total value of each of the measures over the day.

	#Reads	#Writes	#Errors
Sum =	198	175	37

(3) Because this is the initial derivation of the profile, no historical mean yet exists.

	#Reads	#Writes	#Errors
Historical Mean =	0	0	0

(4) The Cross Product Matrix is derived by using the formula:

$$Cprod(x,y) = \sum_{Y^{th}\,meas\,seg.\,i} v_i(x)\,v_i(y)$$

	#Reads	#Writes	#Errors
#Reads	3184	2713	600
#Writes	2713	2555	526
#Errors	600	526	135

The Covariance Matrix was derived using the formula:

$$Cov(x,y) = 1/n\left(\sum_{i=1}^{n} v_i(x)\,v_i(y)\right) - Mean(x)\,Mean(y)$$

	#Reads	#Writes	#Errors
#Reads	132.667	113.042	25.000
#Writes	113.042	106.458	21.917
#Errors	25.000	21.917	5.625

The Inverse Covariance Matrix (C^{-1}):

$$C^{-1} = \begin{bmatrix} 0.010 & -0.075 & -0.156 \\ -0.075 & 0.103 & -0.692 \\ -0.156 & -0.692 & 1.141 \end{bmatrix}$$

The preceding example demonstrated the derivation of the four components of the statistical profile of the use of an object during one day applying the statistical model used in the IDES. The first step was to determine from the measured audit data in Table 3.1 how many complete time periods (in this case, hours) that each of the three measures was observed. In this example, the number of reads, writes and errors was observed for each of the possible 24 time segments during which each of these measures has been observed for this subject. Thus the effective count for each measure is 24. The effective count vector is the first of the four components of the profile. The second step is to sum the values for each of the measures over the number of time segments measured (the count) and determine the mean values for that day. The vector of historical mean values is the second component of the profile. Next, the cross-product matrix, a component of the active data which would be passed up to the global object monitor is derived. The third component of the object profile, the covariance matrix is derived. Note that the values for the mean(x) and mean(y) are both zero for this example because this is the first day of auditing and no historical mean has yet been established. Finally, the fourth component of the object profile, the inverse of the covariance matrix, is determined.

3.4 LAN-IDS

The LAN-IDS is the major component of the two-level Network Monitor Model. It is also the first line of intrusion detection in that it should catch the majority of detectable intrusions. This claim is based on an intuitive assumption that most computer abuses will originate from the same system on which they occur and that these abuses will usually be simpler in that they will involve a single perpetrator acting alone.

Like the IDES, the LAN-IDS exists on a physically separate dedicated workstation which is connected to each LAN target system. It receives encrypted, standard-format multiplexed datagrams from each of the target systems on the LAN. Each datagram contains one audit record which describes a single-object event. The audit record's data becomes one entry in the audit matrix.

Anomaly Detection

After decrypting the datagram, the LAN-IDS has two major tasks. One task is to analyze the audit record to determine if it recorded an anomalous event. The steps involved in this process are given in figure 3.5. The statistical analysis consists of four tests: two with subject profiles and two with object profiles. First, the

anomaly detector performs a statistical test to compare the subject's most recent action, as given by the audit record, with the subject's historical behavior on that machine, as given by the subject's profile for that machine. If the results of this test indicate that the observed behavior was different from the behavior expected at that point, an anomaly is recorded and a warning is issued to the security officer through the administrative interface. If, however, the behavior was within the acceptable range of expected values, no anomaly is recorded and the second layer detection test is applied to the audit record data. The anomaly detector performs a statistical comparison test of the audit record data with the subject's combined profile which portrays the user's aggregate behavior over the LAN This test might detect a user who is distributing portions of an abusive activity over several hosts on the same LAN in an attempt to cover the actions. An example is a user who attempts to browse through other's directories using many change directory commands followed by directory listings and file accesses. The user is clever to limit the number of times he/she does this on each of several machines so as not to attract attention. If this combined subject profile test results in an anomalous finding a warning is sent to the security officer. If nothing especially unusual is noted, the anomaly detector conducts the next test.

The third test of the detection process is a comparison test that involves all of the audit record data, including the anomalous records that were found in the first two tests. This audit data is compared with the appropriate object's profile for the machine involved. This test aims to catch suspicious or abusive usage of an object when that usage has not been revealed by the earlier two subject profile tests. As with the subject profile test, if an anomaly is detected, a warning is issued; otherwise, a second layer final comparison with that object's combined LAN profile is conducted. This test compares the object usage as reported by the audit record data with the expected usage of that object network wide as determined from past usage of that object. This test might detect that although no one user had accessed a classified file an unusual amount of times in the last hour, the cumulative number of accesses far exceeded the norm for that file for that time of day. Therefore, this suspicious activity needs to be investigated further so a warning is issued.

Concurrently with the statistical anomaly detection process, the Expert System component would be applying tests about the known system vulnerabilities or known attack scenarios to the audit record data in an effort to detect any activity which is anomalous on its own accord

independent of the user's past activity. Otherwise, the LAN-IDS assumes that event was not abusive unless notified later by the GOM.

The previous description of the anomaly detection method is effectively a filtering process. See the flow graph given in figure 3.5. The LAN-IDS is expected to discover most of the intrusions in the first comparison with the subject profiles. The combined subject profile comparison will aid in discovering those intrusions enhanced by the networking environment. The object profile and combined object profile comparisons are expected to help in noting the more rare and more complex multi-user or multi-system intrusions.

Profile Updating

The other task of the LAN-IDS upon receipt of an audit record from the target system is to update the profiles. This update process is independent of the anomaly detection process. At the end of each user session (or some other chosen time period) on a particular system, the user's profile is updated using all of the audit records collected for that session on that system. If the user had accessed a remote machine during that session, the audit data would only record the information about the events that connect

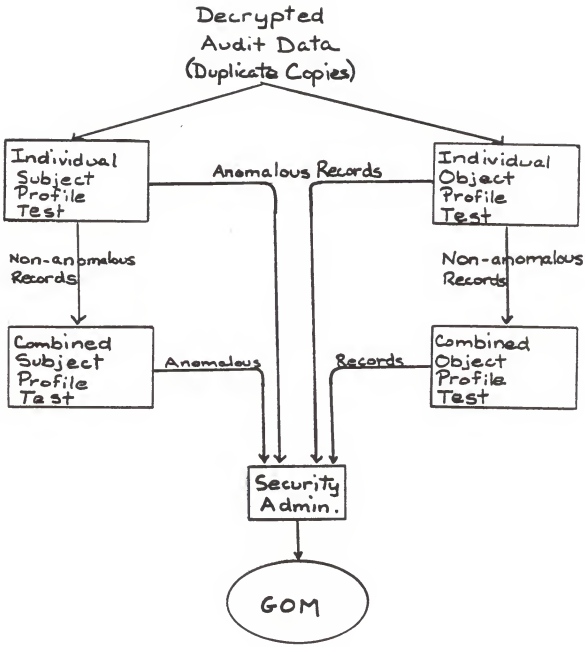

Figure 3.5. The LAN-IDS Anomaly Detection Process.

and disconnect with the remote machine. Any actions the user performed on any objects at the remote machine would be audited at the remote machine and would be used to update the profiles pertaining to that machine. An example is given below:

On machine1 user A does:	On machine2 user A does:
% rlogin *machine2*	> *user_id* > *password* > rm *file1* > logout

The audit records at machine1 would contain such information as the remote login to machine2, the closing of the connection, and the connect time. The audit records at machine2 would contain data such as:

+ login time (remote)
+ login errors
+ origination (machine1) of remote login
+ removal of file1
+ logout (remote)

Of course, this auditing characteristic applies to object profiling as well.

Once the subject's profile is updated, it is combined with that same subject's profiles for each of the other systems of the LAN. As stated earlier, the exact method for combining the profiles would depend upon which statistical model was used to create the individual subject profiles. The combined profile would basically be a weighted average (by a use factor) of the user's behavior on each of the

LAN's systems.

This combined user profile may be useful for detecting a user who attempts to vary his/her profile over time on one machine in order to perform an abuse on that machine at a later date. Unless the conniving user varied his/her behavior on all the LAN's systems to which he/she had access, the planned abuse might be anomalous in comparison with the combined LAN profile.

Unlike subject (user) profiles which are updated at the end of every user's session, all object profiles are updated at regular intervals such as every hour. The update data would come from all the audit records collected which pertain to actions occuring during the past time period (hour) upon a particular object on a particular machine.

Once all objects are updated, their profiles are combined with their corresponding profiles on other target systems in the same manner as the subject profiles.

3.5 GLOBAL OBJECT MONITOR

The goal of the GOM is to add depth to the Network Monitor Model by detecting those intrusions not evident at the LAN-IDS level. Most of the objectives it uses to meet that goal stem from the numerous internetwork communication

capabilities available to the network user. A partial list of these objectives include:

1) To detect collusion between parties configured as listed in section 2.4.

2) To detect multiple masquerade attempts made at different sites and machines on the network.

3) To detect data aggregation resulting from multiple accesses at several separate network locations.

4) To detect users who are able to take advantage of time delays between entry updates to distributed databases, i.e. a savings account holder who withdraws his full account balance from two different bank branches in quick succession before the central database is updated.

Due to the nature of these objectives, they normally would not be expected to occur within a single time period. Therefore, the GOM would not have the same urgency of the LAN-IDS's to detect intrusions in real time with the intent of immediate reaction. Instead, such as in the case of data aggregation, the GOM would build up the 'evidence' over a period of time and would only notice an anomaly after analyzing all the composite profiles it produces from the information it receives from the LAN-IDS's.

The GOM receives most of its information from the LAN-IDS's. The information passed to the GOM from the LAN-IDS's

includes:

1) All audit records received by the LAN-IDS's. These audit records will be used in the GOM in the same way that they are used in the LAN-IDS's -- in statistical comparison tests and in expert system pattern matching.

2) All composite subject profiles derived by the LAN-IDS's. The set of composite subject profiles from a LAN-IDS(x), LCP-S(x) = {CP(s0), CP(s1),...,CP(sk)}, where CP(sk) represents the composite profile for subject k and k is the number of subjects profiled at LAN(x). At the GOM, the set of global subject profiles is the union of all LCP-S(x) sets where x is the number of monitored LAN's in the internet.

3) Composite object profiles derived by the LAN-IDS's as selected by the internetwork controller. These object profiles would be those that pertain to the objects most vulnerable to internetwork intrusion as discussed in section 3.3. An example would be a database which is distributed over a large portion of the network. Because all LAN's may not be controlled by the same authority, the measures they each track for a particular object may not all be the same. In other words, LAN A may track 20 measures about an object O and LAN's B and C may track those same 20 measures plus 4 more. This difference will not affect the merging of the 3 profiles. Object O's global level profile will have 24 time segments. The global object profile would be the union

of all sets LCP-O(x) = {CP(o0),CP(o1),...,CP(ok)} where CP(ok) is the composite profile of Object k of LAN(x) and k is the number of distinct objects profiled at LAN(x).

4) All detected anomalies for both subjects and objects as determined by each LAN-IDS. These detected anomalies would be useful for the Global Object Monitor because they give the GOM more complete imformation about the status of the network.

The LAN-IDS passes this data to the GOM at regular intervals, e.g. during non-peak traffic times. A suggested time interval would be daily, perhaps around midnight. The data forwarded by the LAN-IDS must be encrypted and checksums should be used. This is essential to protect the privacy of the users and the integrity of the data. The connection between the LAN-IDS and the GOM would probably be a public communication subnet (i.e. telephone lines) which is vulnerable to wiretapping. The GOM decrypts all data. Data that can not be decrypted or is not in the standard format once it is decrypted is rejected as unusable and a potential fake introduced to the system by an unauthorized person or process. Otherwise, the data is entered into the GOM database. See Figure 3.6. The remainder of the information the GOM uses is received from a security administrator who inputs any relevant data about users and objects directly to the Expert System Knowledge Base. This

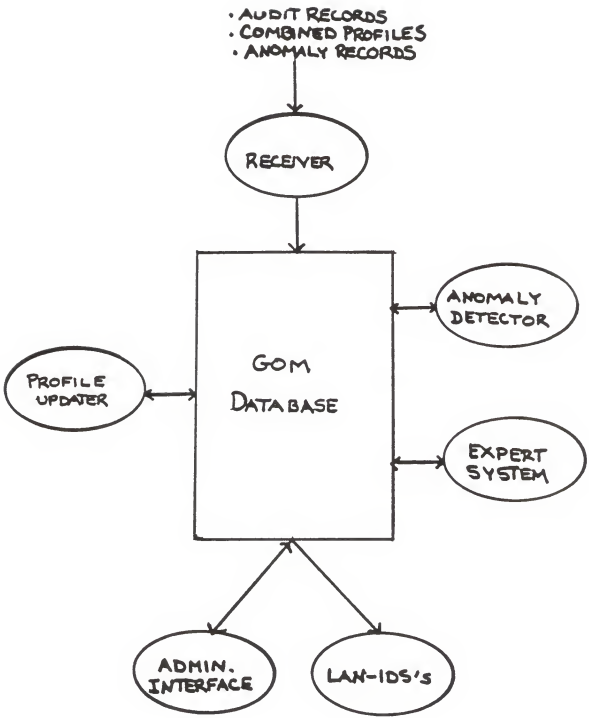

Figure 3.6. The Global Object Monitor Architecture.

data could include information about personnel (user) job relocations, personal history, criminal records, etc. These kind of data can help the expert system and the security officer to determine the nature of a suspicious event.

Anomaly Detection in the GOM

Once the data begins to arrive from the LAN-IDS's, the anomaly detection process retrieves the audit record data from the database. The anomaly detector compares the audit data to the statistical composite object and subject profiles it had derived from the last update. This statistical comparison can be conducted in the same manner as in the LAN-IDS. Any anomalies found are reported to the security officer, used to update the database, reported to the LAN-IDS's and the LAN controllers, and made available to the Expert system.

The GOM's Expert System component plays an extensive role in anomaly detection. The Expert System works with known system vulnerabilities, reported attack scenarios and current intuition about suspicious behavior as does the IDES [Lunt89]. In addition it uses the reported anomalies at the LAN and global levels and the global profiles. For this reason, and because the GOM does not detect intrusions in real time, the Expert System begins processing the data after the Anomaly Detector has completed. Thus, the Expert

System has access to more complete knowledge about past behavior of subjects and use of objects on the network.

The Expert System would use its complete knowledge of the internet users and objects to piece together very complex audit trails about objects and combine this with information about subjects and intuitive attack scenarios, etc. to detect some very complex intrusions. For example, the Expert System might see that:

1) The GOM Anomaly Detector noted an anomalous use of Object O. The chief contributor to that anomaly was the 10% higher Read measure network wide of O for that past day.

2) Each of four users, A, B, C and D, displayed behavior which was nearly anomalous based on their global profiles - they each were at the limit of their thresholds. Further analysis revealed that they each accessed object O slightly more often than usual that day.

3) The knowledge base reveals that although A, B, C, and D each work for different companies at different physical locations, they all were born in Leningrad in the Soviet Union and defected to the United States between 1980 and 1983 while serving as border guards at the Berlin Wall.

Merging these facts with the rules in the rule base using the Inference Engine, the Expert System could determine that there was a strong possibility of a serious intrusion,

perhaps a form of collusion by Soviet spies. The Expert System would notify the security officer that this situation needs immediate investigation and could supply him/her with the reasons that this situation appears suspicious.

Profile Updating

The final task of the GOM is to update its global profiles. It does this by combining the composite profiles it receives from the LAN-IDS's for each particular subject and object. In other words, the GOM would combine all the composite profiles of object A received for that day and use this information to update its global profile for object A. The same procedure would apply to subject profiles.

CHAPTER 4

RESULTS AND CONCLUSIONS

4.0 RESULTS

The major results of this research are summarized below:

1. A general, two level model of intrusion detection was introduced and described for a Wide Area Network or other network of interconnected networks. The first (local) level provides detection capability and monitors activity on the Local Area Networks to the extent that the local level is aware of unusual activity. The second (global) level provides the secondary detection capability by its potential knowledge of all activity on the entire network. The Network Monitor Model solves many of the problems associated with monitoring inter-network communication as described in Chapter 2, sections 2.2 - 2.4.

2. The concept of monitoring objects (files, programs, machines, etc.) as well as subjects (users, processes, etc.) [Ande80] was incorporated into the Network

Monitor Model at both levels. At the local (LAN-IDS) level, object profiling was added for an extra measure of knowledge about the security of the network activity. At the global (GOM) level, object profiling became the primary key to detecting unusual network activity.

3. The concept of layered intrusion detection was introduced in the Network Monitor Model. This approach builds on recent advances in intrusion detection [Lunt88x], [Lunt89] modifying and strengthening it with layers of supplemental analysis. The layering concept is most heavily emphasized in the LAN-IDS with the final layer being the Global Object Monitor. The layering acts as a filtering process which attempts to detect abuses with the minimal effort required while having the capability to apply maximal resources to detect the most complex intrusions. Layered intrusion detection has the advantage of efficient, more complete coverage of network activity.

4. The first level of the model, the LAN-IDS, was based on Denning's Intrusion Detection Model [Denn87] and its prototype, the IDES [Lunt88x]. The major functions of the LAN-IDS were described and further clarified with examples.

5. The concept of a Global Object Monitor, the second level of the Network Monitor Model, was introduced. The purpose and functions of this global monitor were described and examples were given.

6. Some comments were presented regarding the optimal statistical model for conducting the statistical intrusion detection process. Time Series Analysis was suggested as one good approach to modeling the expected behavior of computer system subjects and objects.

4.1 CONCLUSIONS

The research currently being conducted on automated audit trail analysis for intrusion detection is an important addition to computer system security. Intrusion Detection Systems will help to ensure that other security components such as physical security procedures and access control mechanisms are indeed effective and will help to identify those computer abuses such as collusion which presently are not explicitly prevented by security mechanisms. Intrusion detection is an extra layer of defense. When implemented in real-time, the advantages of intrusion detection can be extremely valuable especially when that detection facilitates the ceasing of an attack while it is still in progress.

The most important contribution of this thesis research is the extension of the Intrusion Detection Model to the networking environment. The research of others to date has dealt almost exclusively with single target systems. The research reported here goes beyond single system intrusion detection to provide a flexible, generalized approach to monitoring networks and networks of networks. A two level model emphasizing combined object and subject profiling and analysis is introduced. The strength of the model lies in its ability to determine the complete information about all subjects, objects or actions and use this information to determine the security status of an entire network. As a result of the work done here, there is a basis and general framework for continued research towards an implementable network intrusion detection system.

4.2 AREAS FOR FURTHER RESEARCH

Throughout the investigation of automated audit trail analysis and the subsequent research conducted for this work, several unresolved problems and unexplored ideas were encountered. Each of these points could themselves be the subject of further research.

1. Statistical Anomaly Detection. As noted in section 3.2, there is some question about the best choice of a

statistical model for profiling behavior and determining anomalous activity. Before considering a particular model, a sufficient amount of actual data about user behavior and object usage needs to be gathered in order to determine the actual distribution of that data. Then research should be conducted to determine the effectiveness of a time series analysis for detecting anomalous behavior and usage.

2. One security problem of most network operating systems exists in the remote access facilities. When accessed from a remote site, these systems only know about the identity of that remote site which is directly accessing them. If that remote site is only an intermediary, having been accessed from some other remote site, the ultimate destination system is unaware of this true origination. Thus it can be difficult to track down a devious culprit or to even ascertain the extent of some damages. A possible solution to the situation might be derived using Lamport's Virtual Clock Algorithm [Lamp78]. Applying this algorithm to the events involved in establishing remote connections and transporting data between machines will enable the establishment of a partial ordering of these events [Mizu89]. If this ordering were known for all remote activities throughout a network, it should be possible

to determine the true source of every action. Investigation into the feasibility of this theory should be pursued.

3. Implementation. Before the worthiness of the Network Monitor Model can be established, a prototype must be developed to test the feasibility of the concept.

REFERENCES

[Addi88] K. P. Addison and J.J. Sancho. "Secure Networking at Sun Micro-systems, Inc." Proceedings of the 11th National Computer Security Conference. 17-20 October 1988.

[Ande80] J. P. Anderson. "Computer Security Threat Monitoring and Surveillance." James P. Anderson Co., Fort Washington, Pennsylvania, April, 1980.

[Ande89] J. P. Anderson. James P. Anderson Co., Fort Washington, Pennsylvania. Private correspondence, 17 January 1989.

[Barn83] D. Barnes. "The Provision of Security for User Data on Packet Switched Networks." Proceedings of the 1983 IEEE Symposium on Security and Privacy.

[Cohe84] F. Cohen. "Computer Viruses: Theory and Experiments." Proceedings of the 7th National Computer Security Conference, Gaithersburg, MD, 1984.

[Come88] D. Comer. "Internetworking with TCP/IP: principles, protocols, and architecture." Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1988.

[Clyd87] A. R. Clyde. "Insider Threat Identification Systems." Proceedings of the 10th National Computer Security Conference, Baltimore, MD. 21-24 Sept 1987.

[Denn82] D. E. Denning. "Cryptography and Data Security." Addison-Wesley Publishing Company, Inc., Menlo Park, CA, 1982, reprinted with corrections 1983.

[Denn85] D.E. Denning and P.G. Neumann. "Requirements and Model for IDES - A Real-Time Intrusion Detection Expert System." Technical Report. Computer Science Laboratory, SRI International, Menlo Park, California, 1985.

[Denn87] D. E. Denning. "An Intrusion-Detection Model." IEEE Transactions on Software Engineering, 13-2, p. 222, February, 1987.

[Denn87] D.E. Denning, D.L. Edwards, R. Jagannathan, T.F. Lunt, and P.G. Neumann. "A Prototype IDES - a Real-Time Intrusion Detection Expert System." Technical Report. Computer Science Laboratory, SRI International, Menlo Park, California, 1987.

[DoD85] "Department of Defense Trusted Computer System Evaluation Criteria. " Dept. of Defense, National Computer Security Center, Dec. 1985. DOD 5200.28-STD.

[Gass88] M. Gasser. "Building A Secure Computer System." Van Nostrand Reinhold Company Inc., New York, New York, 1988.

[Glig86] V.D. Gligor, et al. "On the Design and Implementation of Secure Xenix Workstations." Proceedings of the 1986 IEEE Symposium on Security and Privacy, 7-9 April 1986, Oakland, California.

[Hann88] M.E. Hanna, M.M. Sebring, E. Shellhouse, and R.A. Whitehurst. "Expert Systems in Intrusion Detection: A Case Study." Proceedings of the 11th National Computer Security Conference, Baltimore, MD, 17-20 October 1988.

[Hoga88] C.B. Hogan. "Protection Imperfect: The Security of Some Computing Environments." ACM Journal of the SIGOPS, 1 February 1988.

[Lamp78] L.L. Lamport. "Time, Clocks, and the Ordering of Events in a Distributed System." Communications of the ACM, Vol. 21, No. 7, July 1978, pp. 558-565.

[Lamp73] B. W. Lampson. "A Note on the Confinement Problem." Communications of the ACM, Vol. 16, No. 10, October 1973, pp. 613-615.

[Lunt88a] T.F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D.L. Edwards, P.G. Neumann, H.S. Javitz, and A. Valdes. "IDES: The Enhanced Prototype - A Real-Time Intrusion-Detection Expert System." Technical Report. Computer Science Laboratory, SRI International, Menlo Park, California, 1988.

[Lunt88b] T.F. Lunt. "Automated Audit Trail Analysis and Intrusion Detection: A Survey." Proceedings of the 11th National Computer Security Conference, Baltimore, MD, 17-20 October 1988.

[Lunt88c] T.F. Lunt and R. Jagannathan. "A Prototype Real-Time Intrusion-Detection Expert System." Proceedings of the 1988 IEEE Symposium on Security and Privacy, April 1988.

[Lunt89] T. F. Lunt, R. Jagannathan, R. Lee, A. Whitehurst. "Knowledge-Based Intrusion Detection." Proceedings of the 1989 AI Systems in Government Conference, Washington, D.C., March 1989.

[Mans88] D. Mansur. "A Network Security Monitor." Lawrence Livermore National Laboratory, Livermore, California. Notes from the 2nd Intrusion Detection Workshop, Gaithersburg, MD, 21 October 1988.

[McNu89] Sallie A. McNulty. Kansas State University, Department of Statistics. Personal communication, March-April 1989.

[Merr74] The Merriam-Webster Dictionary, Pocket Books, Simon & Schuster, New York, New York, 1974.

[Mizu89] M. Mizuno. Kansas State University, Department of Computing and Information Sciences. Personal communication, January 1989.

[Morr79] R. Morris and K. Thompson. "Password Security: A Case History." Communications of the ACM, Vol. 22, No. 11, November 1979, pp. 594-597.

[Neum88a] P.G. Neumann. "Summary of Computer Abuses and Their Anomaly Detection." Notes from the 1st Intrusion Detection Workshop, Computer Science Laboratory, SRI International, Menlo Park, California, 28-30 March, 1988.

[Neum88b] P.G. Neumann. "What Kinds of Computer Misuses Might We Expect to Detect?" Notes from the 2nd Intrusion Detection Workshop, Gaithersburg, MD, 21 October 1988.

[Neum89] P.G. Neumann. Computer Science Lab, SRI International, Menlo Park, CA. Private correspondence, 18 January 1989.

[Picc87] J. Picciotto. "The Design of an Effective Auditing Subsystem." Proceedings of the 1987 IEEE Symposium on Security and Privacy, 27-29 April 1987, Oakland, California.

[Salt75] J.D. Saltzer and M.D. Schroeder. "The Protection of Information in Computer Systems." Proceedings of the IEEE, Vol. 63, No. 9, March 1975.

[Shum88] R. H. Shumway. "Applied Statistical Time Series Analysis." Prentice Hall, New Jersey, 1988.

[Sibe88] W.O. Sibert. "Auditing in a Distributed System: SunOS MLS Audit Trails." Proceedings of the 11th National Computer Security Conference, Baltimore, MD, 17-20 October 1988.

[Tann88] A. Tanenbaum. "Computer Networks." Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[UNIX86a] UNIX Programmer's Manual, 4.3 Berkeley Software Distribution, University of California, Berkeley, CA, April 1986.

[UNIX86b] UNIX System Manager's Manual, 4.3 Berkeley Software Distribution, University of California, Berkeley, CA, April 1986.

[UNIX86c] UNIX User's Manual, 4.3 Berkeley Software Distribution, University of California, Berkeley, CA, April 1986.

[Vatt78] P.A. Vatter, S.P. Bradley, S.C. Frey, Jr., B.B. Jackson. "Quantitative Methods in Management: Text and Cases." Richard D. Irwin, Inc., Homewood, IL, 1978.

[Vaug88] R.B. Vaughn, Jr. "A Security Architecture for Office Automation Systems." A Doctoral Dissertation, Kansas State University, 1988.

[Walk85] S. T. Walker. "Network Security Overview." Proceedings of the 1985 IEEE Symposium on Security and Privacy, Oakland, CA.

A NETWORK MONITOR MODEL FOR INTRUSION DETECTION

by

MARGARET ELIZABETH WILLIAMS TUBESING

B.S., United States Military Academy, 1982

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

COMPUTER SCIENCE

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

ABSTRACT

This thesis proposes a model for intrusion detection in a networking environment. The model represents an extension of the Intrusion Detection Model of Denning [Denn87]. The intrusion detection methodology is derived from analysis of audit log data and the notion that a computer abuse will generally manifest itself in a departure from normal subject behavior or object usage.

The Network Monitor Model is developed as two distinct levels which constitute a layered approach to intrusion detection. The first level of the model, the LAN-IDS, monitors in real-time the activity of all computer systems which are joined together into a local area network (LAN). The LAN-IDS uses a statistical forecasting model and an expert system to analyze audit records of security relevant events for any abnormal subject behavior or object usage on that LAN. The second level of the model, the Global Object Monitor (GOM), oversees the aggregate activity of multiple connected LAN's. The GOM employs the same general detection process as the LAN-IDS but directs its focus towards any unusual usage of specified security relevant objects. Though not a real-time monitor, the GOM has the advantage of access to all the audit data of the entire network.