NETWORK ATTACKS AND AN ETHERNET-BASED NETWORK SECURITY MONITOR*

L. Todd Heberlein, Gihan V. Dias, Karl N. Levitt, Biswanath Mukherjee, Jeff Wood

Division of Computer Science
Department of Electrical Engineering & Computer Science
University of California, Davis, CA 95616

ABSTRACT

The study of security in computer networks is a rapidly growing area of interest because of the proliferation of networks and the paucity of security measures in most current networks. Since most networks consist of a collection of inter-connected local area networks (LANs), this paper concentrates on the security-related issues in a single broadcast LAN such as Ethernet. Specifically, we formalize various possible network attacks and outline methods of detecting them. Our basic strategy is to develop profiles of usage of network resources and then to compare current usage patterns with the historical profile to determine possible security violations. Thus, our work is similar to the host-based intrusion-detection systems such as SRI's IDES [LUNT88a]. Different from such systems, however, is our use of a hierarchical model to refine the focus of the intrusion-detection mechanism. We also report on the development of our experimental LAN monitor currently under implementation. Several network attacks have been simulated and results on how the monitor has been able to detect these attacks are also analyzed. Initial results demonstrate that many network attacks are detectable with our monitor, although it can surely be defeated. Current work is focusing on the integration of network monitoring with host-based techniques.

I. INTRODUCTION

Network attacks or intrusions such as eavesdropping on information meant for someone else, illegally accessing information remotely, breaking into computers remotely, inserting erroneous information into files, and flooding the network thereby reducing its effective channel capacity are not uncommon. To overcome these problems, several proposals suggest the deployment of *new, secure,* and possibly *closed* systems by using methods that can prevent network attacks, e.g., by using encryption techniques. But we recognize that these solutions will not always be proper because of the tremendous investment already made in the existing infrastructure of *open* data networks, however insecure the latter might be. Furthermore, encryption techniques cannot protect against stolen keys or legitimate users misusing their privileges. Hence, we approach the problem from a

*This work has been supported in part by the Department of Energy and the Lawrence Livermore National Laboratory.

different angle. Specifically, our goal is to develop monitoring techniques that will enable the maintaining of information of *normal* network activity (including those of the network's individual nodes, their users, their offered services, etc.) The monitor will be capable of *observing* current network activity, which, when compared with historical behavior, will enable it to detect in real-time possible security violation on the network – regardless of the network type, organization, and topology. Since our goal is to detect network intrusions, note that we are borrowing some of the basic concepts that have been developed or proposed for non-networked, stand-alone, intrusion-detection systems, e.g., IDES [DENN87, LUNT88a], MIDAS [WHIT87], and others. See [LUNT88b] for a survey of intrusion-detection development efforts.

The focus of our present activity is narrowed to the local environment. In particular, we are developing our concepts for an Ethernet (CSMA/CD) LAN which, because of its broadcast property, enables us to design and test a single secure monitor that has access to all of the network traffic. A prototype LAN security monitor – hereafter referred to as our Network Security Monitor (NSM) – has been in experimental operation for approximately a year, and it is continuously been upgraded as we incorporate into it newer concepts as they emerge. The NSM in its most elementary (lowest) level of operation can measure network utilization and host-to-host activity. But when it suspects a possible intrusion or under the control of a Security Officer, it can also refine its focus on an individual user, a group of users, individual or group(s) of services they are using, etc., in a hierarchical fashion. Probabilistic, rule-based, and mixed approaches are being employed by the monitor, and it raises alarms for the Security Officer upon detecting anomalous behavior. The Security Officer interfaces with the monitor via a user-friendly window system, using which he/she can manually alter (usually refine) the monitor's focus as well. At present, the monitor is being employed to *catch* several simulated network attacks, as we report later in the paper. It is also being installed at LLNL.

II. SYSTEM MODEL

The target system, which needs to be protected from attack, consists of a number of host computers (including devices such as file servers, name servers, printers, etc.) and a LAN through which the hosts are inter-connected. The LAN is assumed to employ a broadcast medium (e.g., Ethernet), and all packets transmitted over the LAN are potentially available to any device connected on the network. The LAN is also assumed to be physically secure, in the sense that an attacker (intruder) will not be able to directly access the network hardware such as the connecting medium (cable) and the network interface at each host. The LAN is connected to the outside world via one or more gateways.

The principal source of attacks is assumed to originate from the outside world and not from a source which already has legitimate access to a host or the LAN. However, an intruder's strategy could be to initially infiltrate a less secure host on the LAN and then utilize this trust as a platform for launching the attack on the ultimate (main) target.

Of course, the most effective way of preventing an attack is to isolate the system from the outside world. However, there are

many environments, which, while requiring that the integrity of the system is protected, need to operate in an open environment, as outlined below. First, the system needs to communicate with systems not controlled by its owners, and such systems, and the communication paths to them, are not necessarily trusted. This communication consists of user data (e.g., mail) and system data (name and file service, authentication, etc.) Second, the system needs to be built using off-the-shelf hardware and software, which may have (known or unknown) security problems. Finally, the system must use existing communications protocols.

III. ATTACKS ON NETWORKED COMPUTERS

The sources of network attacks could be hosts on the LAN, devices connected to the LAN (e.g., wiretaps), and devices outside the LAN connected via a gateway. If the system owners have taken sufficient precautions regarding physical access to the hosts and the LAN, and regarding screening of users authorized to use the system, the remaining point of weakness is from outside the LAN. The targets of attacks could be hosts, the LAN (including bridges and gateways), and resources outside the LAN used by the system or its users. An attacker could be malicious or benign. The attacker could also harm the system inadvertently. The objectives of an attacker could include: access the system "for fun"; use computing resources (CPU, disk, I/O devices, etc.) for his own purposes; obtain information stored on the system; modify or destroy information on the system; prevent or impede normal operation of the system; or damage or destroy the system.

An attack could be considered to comprise of three phases, viz. preparation, execution, and post-attack. In the preparation phase, the attacker gathers information needed to launch the attack. The actual attack occurs in the execution phase. In the post-attack phase, the desired effects (including side effects) of the attack are observable. The three phases are analyzed below.

A. The Preparation Phase

The effectiveness of an attacker, both in term of how far he can penetrate the system and how well he can avoid detection, depends to a large extent on how well-informed he is. The corresponding information is of two types – generic information such as break-in methods, common passwords, and weaknesses in operating systems; and specific information about the system to be attacked such as the number, types and names of hosts, the network configuration, the software (both system and applications) being run, users, their work patterns, and personal information about them (useful for guessing passwords), and information about sensitive data on the system.

A competent attacker is expected to have the generic information. However, he also needs the system-specific information. While there are a number of ways of obtaining such information (phone books, drivers license information, inside contacts, etc.), the network itself is a fruitful source of such information. Some utilities which provide a wealth of information in the Internet environment are: The Domain Naming System, NICname/whois service, Finger, Ruptime/rwho, and Sendmail. Details of these services and how their misuse can be detected are discussed in Section IV.

B. The Attack Phase

Assume an attacker A (a hostile program or a human sitting at a computer). A wishes to attack a target T (such as a host, a service, or the network itself). In order to do so, A must establish a channel of communication with T. This may be done by A and T communicating directly with each other (for purposes of this discussion, a network operating as intended is considered simply as a communications channel and not an intermediary) or via an intermediary I, where A communicates with I and I communicates with T. An example of using an intermediary would be to remotely log in to a machine and then access another machine from it. For example, if a network component such as a gateway were subverted and made to perform differently than intended, then it would be considered an intermediary. In general, there could be n intermediaries, where $n \geq 0$.

Consider such a chain A - I(1) - I(2) - ... - I(n) - T. This implies that the attacker has obtained some measure of control over A and the I's and is using them to launch an attack on T. However, A must have launched an attack on I(n) from A and I(1), I(2), ... I(n-1). Therefore, we see that an attack using a chain of intermediaries can be decomposed into a series of attacks, each of which adds to the set of entities under control of the attacker. For simplicity, we consider A and all of the I's together and refer to the composite group as A. Then the attack simplifies to an attack from A to T, where A is a set of entities rather than a single entity.

For A and T to communicate, T must either *offer a service* which can be exploited by A, or T must seek to *use a service* offered by A. A may get T to use a service controlled by it by either obtaining control over a legitimate service provider or by impersonating one.

(i) Services offered by hosts. The lowest level of service provided over the network by hosts is the receiving and sending of packets. At the Ethernet and IP levels, hosts may accept, reject, or forward packets based on their source and destination addresses, protocol types, and other characteristics such as security options. Examples of higher level services are *remote login, finger* and *network file systems.* Securitywise, services can be ranked on two criteria, viz. the *degree of control* over the system given by the service, and the *strength of the authentication* performed. Ideally, as the degree of control given increases, so should the strength of the authentication.

(ii) Services offered by network. The primary service offered by a network (including gateways, etc.) is the transmission of packets. Other services offered are the routing of packets and response to network management commands. These services too can be ranked based on the degree of control provided and on the authentication required.

(iii) Services used by hosts. Hosts use the services provided by the network to send and receive packets and the services provided by other hosts such as resource location, network file systems, etc. In this case, a host is vulnerable to incorrect information being provided by the service. For example, a resource locator may return the identity of a resource controlled by the attacker. The purpose of authentication in this case is to ensure the legitimacy of the information being provided.

(iv) How attackers may exploit services. An attacker may utilize a service in two ways. First, the service, as documented and intended to operate, may contain security holes and weaknesses. These may be compounded by poor operating practices of users and system administrators, e.g., poor choice of passwords. Second, due to bugs and trapdoors, the implementation of the service may allow attackers to use the service in ways not intended by the designers. (Note that there is sometimes a fine line between bugs and features!) For example, in some operating systems, hitting an interrupt character before the login authentication is completed will allow a login without a password, and some operating systems will crash the host when certain types of Ethernet packets are received. Some other services provide debugging modes which give the penetrator privileged access.

(v) Examples of services. We give examples of some services offered and used by BSD Unix together with what they allow a user to do, and the type of authentication performed.

Service	Allows	Authentication
finger	information about users	none
mail	writing to mail file	essentially none
user FTP	read/write files	password
anonymous FTP	read/write restricted set of files	none
rlogin	log-in privileges	access control list / password
name service	name - address translation	host address
network file systems	read/write/execute files	host address

C. The Post-Attack Phase

A system may continue to exhibit changes even after the activity of the attack is over. This residue of an attack may consist of the effects desired by the attacker and possible side effects. From the point of view of the system owner, effects of an attack could include:

- Dissemination of data stored on the system.
- Loss or reduction of system services, possibly due to the attacker's use of services or by the attacker causing damage to the system.
- Loss of system integrity and confidence in the system. Once a system has been penetrated, there is always a possibility that the attacker may do so again, possibly via trapdoors left open the first time.

IV. DETECTING AN ATTACK

The principal problem in detecting an attack is distinguishing it from normal system activity. Our approach is to rate activities on their likelihood of being an attack and concentrate on those deemed more likely to be an attack. The following criteria are used.

- Source of the message – Some sources, especially those outside the LAN or those with low intrinsic security (e.g., terminal servers, PC's, etc.) are more likely to launch an attack.
- Destination of the message – Both hosts which contain sensitive information (or are otherwise attractive to an attacker) and hosts with poor security (which can be used as

intermediaries) are likely targets of an attack.

- Service used – Services with poor authentication, or services which yield more advantages to the attacker, are more susceptible.
- Contents – The contents of messages can be analyzed to determine their legitimacy. In general, this is hard, since contents of user messages tend to be unstructured and vary widely. Control messages used by various protocols (e.g., mail, the initialization part of rlogin) are well structured, and can be analyzed.

It could be expected that an attacker would attempt to make use of the network services described in Section III to obtain information to prepare for the attack. Therefore, we can detect such attempts by monitoring the network. Because there are many legitimate uses for such information, the majority of queries may not be indicative of an attack. However, excessive queries to such services or queries which appear to be gathering information and which would be useful to an attacker may be an indication that preparation for an attack is in progress.

Accessing system services from unusual locations, at unusual times, or with unusual patterns of activity may also be an indication of an attack in progress. In deciding whether an observed activity is an attack, not only the documented features of the service but also possible bugs must be considered. Unusual or infrequently-used services may be regarded with more suspicion, since well-known services have been extensively analyzed and have stood the test of time so that most of their weaknesses are expected to be known and possibly corrected as well.

Detecting that an attack has occurred by observing its effects can be done in two ways. The first is by analyzing system logs and audit trails for evidence of the attacker's actions, and the second is by observing changes to system behavior due to the attack. Examples of the latter are hosts crashing or not responding to network queries, or sending unusual types or numbers of messages. If sensitive data is tagged or can be otherwise identified by observing it when it is being sent across the network, it is possible to monitor network traffic for such data being read by an attacker.

In the following subsections, examples of some known methods of attack are analyzed.

A. Whois / Finger

The *whois* and *finger* services provide information about users of the system. While the information provided is (or should be) non-sensitive, it could be used for compiling information about an organization (such as which department a person is in and the composition of project groups). It can also be used to gather information on account names and log-in patterns of users in preparation for an attack.

Detection of *finger* attacks is done by observing unusual patterns of activity. For instance, repeated fingering of the same host, or fingering of all the hosts on a network, may be considered suspicious.

B. Mail / SMTP

Mail is typically considered a write-only service. However, it is possible to perform an attack by sending a message which executes system commands when the recipient reads or executes it. Such a message would contain a Trojan horse. In the Internet environment, mail currently does not allow authentication (although use of RFC 1113-1115 may change this) and it may be possible to get a mail recipient to take some action by sending forged mail. Also, excessive use of mail could flood the system or overflow the recipient's mailbox.

The SMTP (Simple Mail Transfer Protocol) service performs other actions in addition to delivering mail. It can be used to ascertain the contents of mailing lists and verify user id's. Also, if its debug mode is enabled, it can be used to issue commands to the system.

Mail-based attacks can be detected by comparing the Internet source address with the source specified, and by monitoring the SMTP service to ensure that it is followed correctly. Also, mail connections can be monitored to ensure that only authorized gateways send and receive mail from outside the organization.

C. Remote Login

Remote login, as provided by the BSD Unix *rlogin* service or similar services such as *rsh* or *telnet,* allows a user to give commands as if he were on a directly connected terminal, and it enables users to run arbitrary programs on a system from a remote location. Many systems allow system administrator (root) privileges by remote login, and for good reasons ... to allow remote debugging of a system. The *rlogin* command allows access to a system by giving a password or by being on an access control list (ACL) kept on the system. The ACL is a list of host and user names. In addition to logging in by ascertaining the password, an attacker could use the access control facility by

- Obtaining access to an account listed in the target's ACL.
- Obtaining access to the file the list is stored in (*.rhosts*) and modifying it.
- Subverting the name to address translation service to yield a false address for a host in the ACL.
- Subverting the network to make it appear the attacker is at a host address corresponding to a host named in the ACL.
- Obtaining system privileges on a host named in the ACL and masquerading as the user named in the ACL.

The above illustrates the fact that security is obtained from a chain of factors and the chain is only as strong as its weakest link. In normal use, remote logins tend to be from local hosts plus a limited number of remote hosts. Monitoring of the source address and the destination host and user is useful in detecting login attacks.

D. Network File Systems

Network file systems are services provided by hosts to other hosts on the network, and hosts are therefore susceptible to attack both as service providers and service users. In the first case, an attacker could access a network file service to read from and write to files, and in the second, he could provide a file system containing bad data or programs to be used by the system under attack. As in the case of login, detecting file system traffic

to or from the outside could be a sign of an attack.

E. Misrouting Network Traffic

Network traffic can be misrouted by sending fraudulent control messages to gateways, routers and hosts. Misrouting could be used to direct unauthorized traffic to an attacker's machine, where it could be examined and possibly modified, to masquerade as hosts, or to prevent hosts from communicating with each other. Attacks on network components can be detected by observing control packets used to modify routing tables, and by comparing routing tables in hosts and gateways with the expected values.

F. Overloading the System

A denial-of-service attack can be performed by overloading various parts of the system, such as hosts, the network, and gateways. One method of overloading a network is by simply sending a stream of packets which exceed the capacity of the network or a segment of the network. Variations of this scheme, in which each packet sent causes a large number of packets to be generated at the target network, can be used to disrupt a network without overloading the sender. Another strategy could be to open a large number of connections to a target host or network. Since host and gateway routing tables have limited capacity, this activity will eventually lead to bona-fide users being unable to connect since all the resources are being used by bogus connections. This type of attack is easy to detect since it results in a sharp increase in network traffic, either as a whole, or from/to a host or subnet.

G. Domain Name Service (DNS)

This is an example of an attack in which an attacker gains control of a service used by hosts in the system. The DNS is used to translate host names into addresses. An attacker impersonating a remote name server could return fraudulent addresses to name queries, thus leading hosts to connect with machines (possibly controlled by the attacker) other than those expected.

H. Eavesdropping

Eavesdropping on network traffic can be done from a device connected to the network medium. Eavesdropping on connections outside the organization is often easy because communication links are often not under their control. As well as obtaining sensitive data, eavesdropping can be used to obtain passwords which can later be used to log in to hosts.

Detection of passive eavesdropping is difficult, unless the physical equipment used for eavesdropping can be located. Physical security and encryption of data sent on insecure links can help prevent eavesdropping. This attack can be detected via active testing, viz. by determining if fraudulent messages are being returned by a host (under the attacker's control).

V. CONCEPT OF THE N.S.M.

This section presents the conceptual view of our Ethernet-based NSM. Currently the NSM uses a four dimensional matrix of which the axes are: Source (a host which generates traffic), Destination (a host to which traffic is destined), Service (mail, login, etc.), and Connection ID (a unique identifier for a specific

connection). Each cell in the matrix represents a unique connection on the network from a source host to a destination host by a specific service. This matrix is similar in concept to the well-known access matrix, the basis for protection in many systems. Each cell holds two values: the number of packets passed on the connection for a certain time interval, and the sum of the data carried by those packets. An analyzer must examine the data patterns in the matrix representing the current traffic to determine if an attack is occurring on the system.

One method to examine the traffic matrix is to compare it against a matrix holding a certain pattern. For example, a comparison may be made against a matrix holding the representation of a specific attack. To compare the two matrices, the pattern being checked can be treated as a mask, and the current traffic pattern can be passed through that mask. Data passing through the mask should be brought to the attention to a Security Officer.

Designing patterns for all possible attacks is difficult at best and computationally infeasible. Therefore, the NSM generates a mask of normal traffic flow, and an inverse (as in a photographic inverse) is made of this normal traffic matrix. This new mask represents all traffic flow outside the normal traffic flow. The matrix for the current traffic flow is passed through this "abnormal" mask, and any data passing through is presented of the Security Officer.

Unfortunately the matrices for the network traffic are potentially enormous, especially if a larger dimensional matrix is considered. Even sparse matrix implementations contain a very large number of cells. Checking each cell against the mask may require more resources than available. The NSM, therefore, groups cells in a logical and hierarchical fashion. The groups are then presented to a mask, which in turn has been grouped. If a group passes through the mask, this group can be presented to the Security Officer; furthermore, the NSM can break the group into the smaller constituents to perform a more detailed analysis.

This hierarchical structuring allows for a monte carlo divide and conquer search of the entire network traffic. If processing power is available, greater analysis may be conducted on groups which do not show abnormality to reduce chances that the probabilistic search presented an incorrect answer.

The second method to examine the current traffic matrix is to apply a set of rules against the matrix. This method is particularly important if profile masks have yet to be generated. Since the rules look for specific traffic patterns, they can be transformed into matrix masks too; therefore, only the single analysis tool, passing current traffic through masks, needs to be used. Unfortunately, after examining a number of potential rules, we have found not all rules apply well at all grouping levels, so a mask may only be applicable at a single level. For example, a rule looking for a login connection which only exchanges a few packets and terminates (thus indicating a possible failed login) does not map well to the Source-Destination group level. Conversely, a rule looking for a host communicating with a large number of other hosts works well at the Source-Destination level, but it does not work well at the connection level.

Details of the NSM can be found in [HDLM90].

VI. PERFORMANCE OF THE N.S.M.

The first analysis of the NSM's performance only included the rule base detection. Probabilistic detection testing will begin shortly; however, a computation of the actual data path space used has been calculated and will be described later. The NSM was tested for twenty days on the our Ethernet. Included in the test data were two simulated attacks. The attacker and the individual writing the rules did not discuss the attacks.

As mentioned previously, only a few simple rules are used; however, these rules have proven useful. The following are the rules used:

- If the total number of connections by a single service between two machines is greater than fifteen, then report.
- If a host communicates with more than fifteen other hosts using the telnet or login services, then report.
- If a connection is attempted to a nonexistent host, then report.

A total of 86 warnings, or approximately four warnings a day, were issued. Most of these warnings were generated by workstations running a large number of X window tools on a remote CPU and by mail connections from our central mail host. Some of the more interesting warnings are described below.

1) A number of hosts copied a large number of files via the File Transfer Protocol. File transfers of over three hundred files were observed more than once. Further investigation showed that people were backing up their files to other machines using ftp. At least one large ftp was to a host at Stanford containing a large number of public domain programs for personal computers.

2) Over three hundred fingers were initiated between two machines. The finger program is one of several services which provide information about the state of a system: who is currently using the system, which people have never logged in (and thus may have a default password), who has unread mail, etc. This information can be used to prepare for an actual attack, so the report caused some concern. Fortunately it was only one of our colleagues launching a simulated attack.

3) Over 150 mail messages were exchanged between two machines which normally do not exchange mail. Further analysis of the system files on one of the machines indicated that the mail was exchanged between only two users – one of them being root on one machine. Further investigations will continue.

4) On several occasions, over thirty login failures were recorded between a dial-up port and a host on our network but not under our control. Investigations are still continuing for these events as well.

5) Several warning were issued concerning a large number of connections made through an unknown service by several new HP workstations. The unknown service appears to be local to the new machines.

6) Finally, a computer game called "Empire" was initiated on a local host, and the game and Internet address were announced on the usenet network. Frequent warnings were issued concerning the number of hosts, often over twenty, which were

communicating with our local machine.

Although the probabilistic detection scheme (passing the traffic through masks), has not been tested yet, the actual data paths and the potential data paths have been measured. A data path is defined to be a means by which two hosts can communicate. This is generally provided by network services on the hosts – communication via removable media such as disks or tapes is not considered. Thus the total number of data paths between two hosts is defined to be the total number of network services by which the two hosts may communicate. The total number of possible data paths is then the number of host pairs possible multiplied by the number of services used.

A data path is considered to be used if at least one connection, on the average, is made on that path every two weeks. A calculation of the total number of data paths actually used on our Ethernet was 0.6% of total number of data paths possible. Therefore our sparse matrix represents only 0.6% of the potential matrix size, and the probability of a random network attack occurring on one of the normally used data paths is only 0.6%.

VII. CONCLUSION AND FUTURE WORK

We have discussed an approach to obtaining network security based on capturing and analyzing network activity. The need for a security monitor is clear: most networks are intrinsically insecure as are the hosts that are attached to the network, and the network must be protected against users (insiders and outsiders) misusing privileges.

The paper establishes an implemented framework (called the NSM) for coping with network attacks. The NSM, working on an Ethernet although most of the system is independent of the network type, captures and analyzes every packet. A use of the network is considered suspicious if it is very dissimilar to previous uses (aka profiles) or is inconsistent with one or more policies. Similar methods for flagging attacks are the basis for host-based security monitors.

The network model offers the opportunity for a hierarchical analysis of activity. At the lowest level, host-to-host activity is analyzed; at the next level it is services, and at the next level it is connections. The lowest level is the first line of defense, passing suspicious behavior to the higher levels. This is the manner in which the NSM works autonomously. Under Security Officer control, the requests for data start at the top level and proceed downward. Work is in progress on a more detailed analysis of network activity involving users and applications.

The paper also presents a model of network-based attacks, the model reflecting the phases of an attack, the services used, and the purpose of the attack. We have used this model to generate trial attacks on the network and to determine the effectiveness of the NSM in detecting such attacks. The attacks have a commonality, in that a user gains access to the network and then attempts to determine what the hosts can offer him or attempts to damage the network. The attacks we generated all involve noticeable increases in activity at one of the three levels of our analysis hierarchy, and were easily detected by the NSM.

Many attacks will take this form, and will be detectable by the NSM in real-time. More subtle attacks will not leave so

obvious a trail in network behavior. For example, an attacker could guess a password for a host, and use the *rcp* facility to copy the password file from another host for the ultimate purpose of cracking passwords. (Of course, the NSM could contain rules to be suspicious of the password file being transferred, but one could easily think of file names that would not be suspicious to the NSM.) Thus a comprehensive monitor would also involve host-based monitors to watch over the activities of individual hosts. We are considering such hybrid systems.

Our initial results with synthetic attacks are promising and the overall framework for network monitoring allows integrating the NSM with the analysis software that is part of current host-based monitors. Clearly, however, it is essential to install the NSM (and other monitors) into real settings for extended times and determine their effectiveness in coping with real attacks.

Finally, we remark that our present network monitoring activities are confined to the local environment because the broadcast property of LANs enables us to design and test a single secure monitor that has access to all of the network traffic. Distributed monitoring of wide area networks will undoubtedly be more complex, and it will be taken up after our experience from LAN monitoring matures. In an irregular-structured, store-and-forward network, a single location of the monitor will no longer suffice since all network packets will not necessarily be routed through a particular node. Hence, the network monitoring functions have to be distributed among several nodes. These nodes will exchange information to reach a consensus on whether an attack is in progress. Noting that some of these nodes might have themselves been compromised, the distributed monitoring mechanism is expected to borrow some of the concepts from the Byzantine generals Problem.

VIII. ACKNOWLEDGEMENT

We thank Mr. Doug Mansur of LLNL for his encouragement and support of this work.

REFERENCES

[DENN87] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engg.*, vol. SE-13, pp. 222-232, Feb. 1987.

[HDLM90] L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A network security monitor," *Proc., IEEE 1990 Symposium on Security and Privacy*, Oakland, CA, May 1990, to appear.

[LUNT88a] T. F. Lunt, et. al., "IDES: The enhanced prototype," Technical Report No. SRI-CSL-88-12, SRI International, Menlo Park, CA, Oct. 1988.

[LUNT88b] T. F. Lunt, "Automated audit trail analysis and intrusion detection: A survey," *Proc., 11th National Computer Security Conf.*, Baltimore, MD, Oct. 1988.

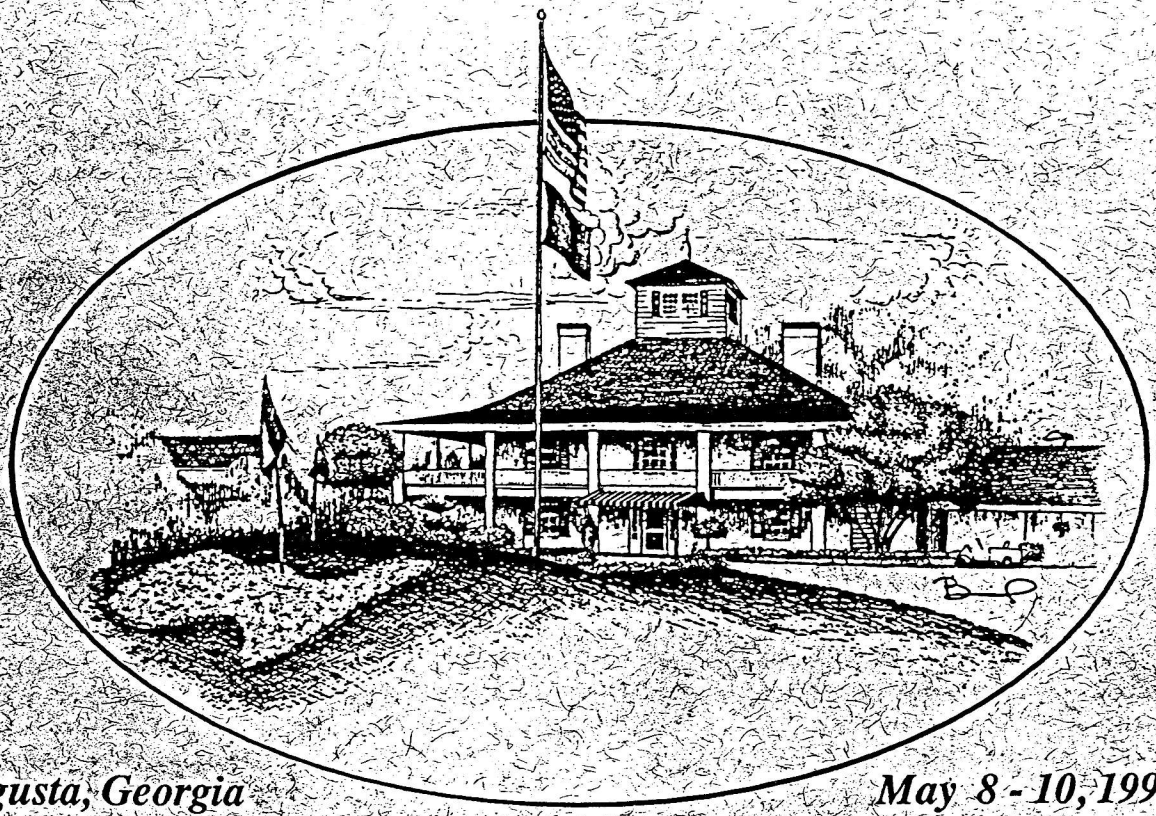
[WHIT87] R. A. Whitehurst, "Expert systems in intrusion detection: A case study," Computer Science Lab., SRI International, Menlo Park, CA, Nov. 1987.

Lab copy

-9005106

13th Department of Energy Computer Security Group Conference

Proceedings



Augusta, Georgia ***May 8 - 10, 1990***

Savannah River Site
Westinghouse Savannah River Company

"Security through Teamwork"

U. S. Department of Energy
Office of Administration and Human Resource Management
Office of Information Resources Management Policy, Plans, and Oversight
and
Assistant Secretary for Defense Programs
Office of Safeguards and Security