Conf-921108--1

LA-UR-92-3214

Title: NADIR: A PROTYTYPE SYSTEM FOR DETECTING NETWORK AND FILE SYSTEM ABUSE

LA-UR--92-3214

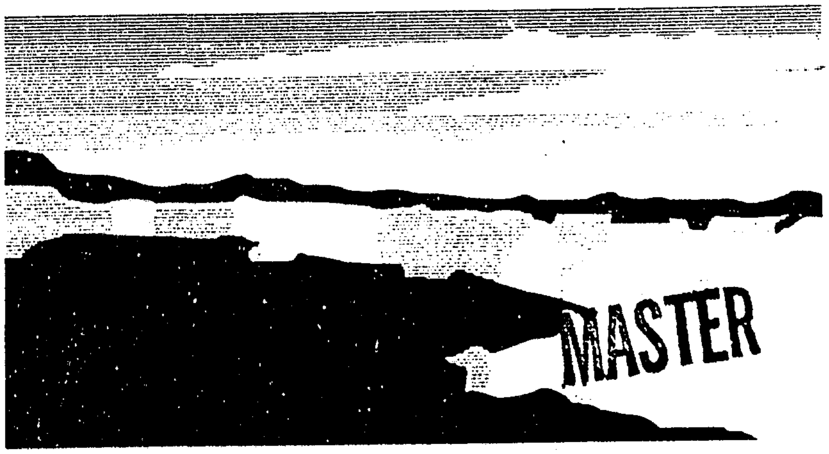
DE93 000861

Author(s): J. G. Hochberg, K. A. Jackson, C. A. Stallings, J. F. McClary, D. H. DuBois, and J. R. Ford

Submitted to: Information Systems Security, Audit, and Control, Brussels, Belgium
November 16-18, 1992

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



MASTER

LOS ALAMOS NATIONAL LABORATORY
3 9338 00819 9480

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NADIR: A prototype system for detecting network and file system abuse[*]

Judith G. Hochberg, Kathleen A. Jackson, Cathy A. Stallings,
J. F. McClary, David H. DuBois, and Josephine R. Ford

Computing and Communications Division, MS B252
Los Alamos National Laboratory
Los Alamos, New Mexico, 87545 U. S. A.

Abstract

This paper describes the design of a prototype computer misuse detection system for the Los Alamos National Laboratory's Integrated Computing Network (ICN). This automated expert system, the Network Anomaly Detection and Intrusion Reporter (NADIR), streamlines and supplements the manual audit record review traditionally performed by security auditors. NADIR compares network activity, as summarized in weekly profiles of individual users and the ICN as a whole, against expert rules that define security policy, improper or suspicious behavior, and normal user activity. NADIR reports suspicious behavior to security auditors and provides tools to aid in follow-up investigations. This paper describes analysis by NADIR of two types of ICN activity: user authentication and access control, and mass file storage. It highlights system design issues of data handling, exploiting existing auditing systems, and performing audit analysis at the network level.

1. Introduction

1.1 The problem

The goal of computer misuse and intrusion detection systems is to discover security violations perpetrated either by insiders (authorized users) or by outsiders who clandestinely enter a facility. These violations fall into four general categories:

- **Disclosure** is the provision of information to unauthorized individuals or organizations.

- **Integrity Violation** is the deletion or modification of data or software. Modified data might be completely unusable, or can be timed to self-destruct at an inopportune moment. More subtle modification can produce a chain of errors in any work based on the data. A major consequence of violation of integrity is a loss of confidence in all data on a system. Violation of integrity can be either an end, or the means by which a computer criminal pursues material gain.

- **Denial of service** is the rendering of a system temporarily or permanently unusable. This can be accomplished by overloading a system, destroying crucial data, or by physical sabotage.

- **Unauthorized access** is the unauthorized use of a system. An outsider can break into the system. An insider can masquerade as another user (usually with higher or different privileges), or can allow access to the system by compromising the system's defense mechanisms. These activities usually lead to further security violations.

[*] The Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36. This work was performed under the auspices of the United States Department of Energy.

The first line of defense against all such violations is the institution of formality of operations: a way of doing business that emphasizes safeguards and accountability. While formality of operations includes institutional practices such as personnel education and physical security measures, our main concern is software measures that restrict access to the computing system and the files it holds. Several factors limit the efficacy of these measures. The first is human nature. Users often see even the most rudimentary security procedures as unwelcome diversions from the main thrust of their work, and therefore fail to apply them. Second, system managers must effect a compromise between the compartmentalization that security requires, and the access to distributed resources that users need if they are to exchange ideas and data with other users within and without the system. Third, systems frequently contain undetected vulnerabilities.[1] Finally, there is the more general threat of the privileged and malicious insider [10].

Given these weaknesses, a second line of defence against abuse is the maintenance and analysis of an audit record of system activity and file system usage. In theory, one can detect break-in attempts, violations of integrity, and denials of service by abnormal or invalid machine activity. However, the traditional approach of manual audit analysis has generally proved unworkable. Human data processing limitations restrict manual review to a sampling or cursory scanning of the large quantity of audit data typically generated. Only a few obvious intrusion or misuse scenarios can be targeted; even these may be missed because of human error.

The limitations of manual review have long been apparent to security personnel at Los Alamos. While manual review by security auditors did reveal many instances of misuse, there was no way to evaluate the general success or completeness of this effort. Large-scale manual audits of past data in response to specific inquiries from the Laboratory's Internal Security office also proved cumbersome and time-consuming. It was obvious that an automated audit record review would be more effective. Such an analysis can combine expert knowledge of security problems with a computer's capability to process accurately and correlate large quantities of data. In addition, the speed of machine processing can allow analysis to be performed in near realtime[2] so that auditors can be notified of suspicious activity quickly, and take action to trace and stop it.

1.2 Target system

The Integrated Computing Network (ICN) is Los Alamos National Laboratory's main computer network. Serving nearly 9,000 users, it includes six Cray-class supercomputers and many smaller computers, file storage devices, network services, local and remote terminals, and data communication interfaces. Through the ICN, any user inside the Laboratory may access any host computer from office workstations or terminals, if authorized to do so and using an approved access path. Outside users typically access the ICN through telephone modems, leased lines, or one of many world-wide networks.

[1] A specific example is the Unix "finger" command, which queries a computer about a user's identify and logon status. For many years a bug in this command treated text beyond the allowable input length as a direct command to the computer queried, enabling any user on a network to circumvent formal access barriers and execute a command on any other Unix computer in the network [21].

[2] In less than 30 seconds.

The ICN is structured uniquely in that it is divided into four "partitions" dedicated to specific levels of processing. Each computer in the network resides in only one partition.[3] The partitions are linked by a system of dedicated ICN nodes that perform useful network services while enforcing the partitioning itself. Tasks performed by these *service nodes* include user authentication, access control, job scheduling, file access and storage, file movement between partitions, and hardcopy output. All service node activities are sensitive to partitioning. For example, they block unclassified users from accessing classified files, and prevent users with unclassified passwords from logging onto machines in the classified partition.

1.3 Comparison with other systems

Automatic intrusion and misuse detection systems typically use one or both of two methodologies. *Anomaly detection* systems compile statistics of normal behavior, then note deviations from these norms. *Expert systems* identify instances of specific misuse scenarios, using rules derived from interviews with security experts and hands-on examinations of audit record data.

The first Los Alamos intrusion and misuse detection system was developed in 1983-84 [1]. It used an expert system to check the audit record for a small set of activities, or combination of activities, which were suspicious enough to raise concern. At that time no body of knowledge on the nature of attempts to penetrate a computer system similar to that at Los Alamos existed. The rules defining significant events were few, and were more conjectural than empirical in origin. Even so, the results of this program development were encouraging.

The pioneering research of Dorothy Denning and her colleagues, and the IDES[4] research and development project at SRI International, heavily influenced further development at Los Alamos. Denning proposed monitoring standard operations on an operating system for deviations in usage -- an anomaly detection approach. Her early research tried to define the activities and statistical measures best suited to do this [3, 4], and continued with the development of the IDES prototype [5]. Since then, Teresa Lunt and her colleagues have continued to develop the IDES system [15, 16, 18]. They have expanded the original concept by adding an expert system component that addresses known or suspected security flaws in the target system. Still, anomaly detection remains the primary emphasis of the IDES approach, with the expert system used to catch invalid activities missed by the first means [17]. The Denning model is the core of many, if not most, intrusion detection systems [6, 14, 19, 22].

Most audit record analysis research, including Denning's, focuses on finding security violations on single mainframe computers or workstations. Our research effort differs in a key respect: whereas that work targets specific systems, ours addresses the security of a *network* connecting many systems. From the security perspective, these domains are both similar and different. Standard operations on a single system (logon, program execution, file and device access) have network-level analogues (authentication and access control, job control, and file access and storage). They differ in that network operations are distributed rather than concentrated on one machine. Nonetheless, if we view our network as one large distributed op-

[3] More accurately, each port into the ICN is assigned a partition; any workstation or terminal connected to that port can access computers at that partition or below.

[4] Intrusion Detection Expert System.

erating system, then the Denning model applies well to the problem of network intrusion and misuse detection.

NADIR [12, 13] analyzes network-level activity via the audit records generated by the ICN service nodes. These records have long been required for accounting and security purposes. It was straightforward to make use of them as input for our system. Because the service nodes record activity (but not network traffic) at the service level, we kept the quantity of data to be processed to a manageable level. Also, because the service nodes play such a critical role within the ICN, we obtained data sufficiently detailed for an effective detection system. Other current network-level efforts either target network traffic at the service and protocol levels [7, 8, 9, 20], or collect data from network hosts for processing by a centralized system [23, 24].

In making the shift from single-system to network security we encountered a host of new challenges. Combining data from across the network forced us to design a system that was both modular and integrated. Individual sources of data had to be taken on and off-line easily. Simultaneously, data from all sources had to be collated and analyzed. More generally, this shift forced us to tackle basic problems of handling and analyzing a large and diverse database. We had to reduce data to summary profiles, detect and eliminate errors, and present results efficiently. In addition, we had to deal with the standard problems faced in imposing an audit analysis system onto a pre-existing and complex system without disrupting the normal conduct of business. Because we believe these issues to be of general interest to the security community, and to the larger community of data analysts, we devote section 2 to a discussion of them. Section 3 describes our methodology, focusing on our specific solutions to these problems. Section 4 contains our results, and section 5 our conclusions. We also touch on the possibility of developing general tools to address the problems raised, and suggest other areas, such as fraud detection, to which our work might be applied.

2. General Issues

2.1 Analysis of an established computing system

Like most designers of misuse and intrusion detection systems, we did not have the luxury of shaping our target to our detection system. The design process went in the opposite direction: we adapted our methodology to the current ICN. To simplify this process we used, for the most part, existing audit, storage, and data transmission capabilities. Current audit record maintenance at the Los Alamos computer facility was a plus. We kept target system impact to a minimum; they had only to reformat the audit records, where necessary, and transmit them to NADIR.

Besides security enhancement, our work has benefited the ICN in three ways:

- *Error Detection.* We uncovered and corrected errors in both the systems audited and in the auditing process.

- *System performance evaluation.* The software we developed for summarizing and analyzing audit data can be used from both a performance and security perspective. We now routinely use our capabilities to help learn more about the everyday operation of ICN systems.

- *Emergencies.* Catastrophic programming errors can result in severe disruption of network services. If detected during execution, operator intervention can minimize their impact. In addition, we use our capabilities for post-hoc analysis, thus helping to prevent recurrences.

2.2 The problem of multiple data sources

NADIR draws on data from several ICN service nodes. Dealing with these multiple sources of data raised design issues of modularity and data transmission:

- **Modularity** has better enabled us to deal with differences among our multiple data sources. The various service nodes make use of several hardware configurations and operating systems. Their software is written in several different programming languages and has been subject to many changes and upgrades. The audit records generated by the service nodes differ substantially from each other in content and in format. We resolved these differences by writing separate software for bringing in data from each node. Modularity also allows NADIR to function properly while accessing data from any subset of its targeted service nodes. This has been necessary throughout the system development process, as we integrated different nodes into NADIR at different times. It will allow future nodes to be brought into the system with a minimum of effort, as current nodes will be unaffected. In addition, it enables individual service nodes to be temporarily disconnected from NADIR for maintenance, modification, or because of failure.

- **Data transmission** has been a necessary step, as obtaining a complete picture of ICN usage required integrating data from all service nodes into one common database. This was also more efficient than implementing separate analysis systems on each service node. Transmission has proved to be a thorny problem because of the large quantities of data involved. While SAM usage is light, with around 200 events per week, the NSC generates close to one megabyte of data per day, and the CFS generates four to five megabytes.

2.3 Handling large databases

We confronted many problems endemic to the general task of analyzing large databases. As these are mundane problems, such as cleaning up and simplifying databases, they have received much less attention in the intrusion detection literature than the more glamorous area of data analysis. Nevertheless, they easily occupy most of the researcher's time, as software for handling these tasks is normally written specifically for each application. We expect in the next few years to see general tools developed to streamline this part of the research process.[5] Such tools will be increasingly important as databases roughly hundreds of terabytes per day, such as satellite data, become ready for analysis.

In the following paragraphs we briefly describe and exemplify several data handling issues. We hope that this will benefit other researchers working with large databases, and encourage the development of more general solutions. We use the term *record* to refer to an entry in an audit record (e.g., an attempted logon; this corresponds to a row in a standard spreadsheet). *Field* refers to something measured for each record (this corresponds to a standard column). *Value* refers to an individual item in a field (e.g., *Maria* in a field containing first names).[6] Some of our terminology and examples are particular to databases of audit logs, though the issues are more general than that.

[5] It is encouraging to note that the National Science Foundation has a new interdisciplinary program for research on scientific databases.

[6] Other common terms are *case, message,* or *transaction* for record and *variable* or *feature* for field.

- **Data parsing** is more than a mechanical task. To parse the incoming data stream one must understand how records are demarcated, how fields are demarcated within records, the format of each field, and the possible range of values within each field. Field formats must be compatible with the allowable formats in whatever system the analyst is using (database manager or statistical software). Three factors can complicate this process:

  Lack of information. Clear documentation of auditing systems is often lacking, and personnel in charge of the systems cannot always provide complete explanations. The worst-case scenario in data parsing is 'legacy code' -- a program in long use for which the source code and the original programmers are unavailable.

  Headers. In some auditing systems, records are periodically written into a computer file whose header gives some general characterization of the records within the file, e.g., the date they were generated. This header also must be understood and parsed, especially if the analyst uses it to generate additional fields for the data as records are read into the database.

  Idiosyncratic organization. Not all data sources are neatly organized into records and fields, although most can be rearranged to fit that model. Sometimes data that one would want to interpret as belonging to a single record are dispersed among several lines, often separated by data from other records. Not all information might be present for all records, and the pattern of dispersion might differ for different record types. Then detailed programs must be written to locate and unite a record's component parts.

- **Data rationalization** refers to the process of identifying and dealing with differences between expected and actual characteristics of the data. The most obvious part of this process is the detection and correction or elimination of data errors. Entire fields may turn out to be meaningless, e.g., all zeroes. A record may be defective, lacking values on one or more key fields. It may have a value that is beyond the range of values defined for a field.[7] It may contain an impermissible combination of values on two or more related fields. Errors can be handled by deleting erroneous records and meaningless fields, and by entering a 'missing value' symbol for individual erroneous values. Persistent error patterns can sometimes be eliminated at the source in consultation with auditing system managers. Often, apparent error patterns turn out not to be errors at all, but misunderstood features of the auditing system. Once these patterns are understood, in consultation with system managers, they can often be rationalized by transforming the data into a less idiosyncratic form. We have identified four factors that tend to make a data source harder to rationalize:

  Age. Newer systems tend to be cleaner and more suited to current analytic needs.

  Modification history. Auditing systems that have been modified tend to be harder to rationalize. Modified systems are often more complex. New fields are added and new possible values are added to existing fields. Modification also introduces an opportunity for errors to enter the system, and for documentation and practice to diverge.

[7] Erroneous values are distinct from values which are permitted but anomalous, e.g., an unusual temperature reading which is nevertheless within a temperature field's defined range.

Target system complexity. Audit records for complex systems are themselves more complex. They have more room for error and confusion and therefore normally require more rationalization than simpler data sources.

Reference. Records are always generated with respect to some unit of activity, e.g., user or computer actions. Rationalization is complicated when there is a mismatch between the audit log's perspective and that imposed by the analytic task.

- **Data fusion** refers to the process of reconciling differences between related data from different sources. Information contained a single field of one data source may be spread out over two or more fields in another data source. Users may be identified by numbers in one data source and by name in another. Fields that are similar in content may be different in format.

- **Data aggregation** is the process of grouping ordinal or categorical values within a field into a small set of categories. This process is useful when a field contains more information than required for a given analysis. For example, temperatures can be aggregated into a few categories such as *hot, warm,* and *cold.*

- **Data reduction** minimizes the dimensionality of a database by compressing fields or records. Data reduction helps keep data storage to a manageable level, and often makes data easier to interpret. Data can be reduced across fields by combining several fields into one superordinate field. For example, values from machine type, operating system, and security level can be used to derive a categorical field of 'computing environment'. Such a process demands that the analyst define all possible categories of the derived variable. Within fields, data can be compressed by profiling. Profiling summarizes audit data across a set of records, usually selected by some unit of time (e.g., daily or weekly profiles). Profiling is always done in reference to one field, according the analyst's specific interest. For example, system audit data can be profiled by user ID or by machine type, with the former appropriate for intrusion detection and the latter for performance analysis.

3. Method

NADIR is implemented on a SUN SPARCstation II[8] running the Sybase[9] relational database management system. Service node audit records are read into NADIR daily. NADIR summarizes these raw data into weekly profiles for individual users and composite profiles for the ICN as a whole. Expert rules, developed through data analysis and consultation with security experts, are applied to the profiles, with weights assigned for each rule triggered. Weekly reports describe overall network usage graphically and numerically, and highlight the most suspicious users identified through the expert rules. Additional reports on raw or profiled data also can be generated on demand, and tailored to the analyst's specific queries.

3.1 Data

To date we have incorporated audit data from three service nodes into the NADIR system. The Network Security Controller (NSC) provides for authentication and access control on

[8] SUN SPARCstation and SUN workstation are trademarks of SUN Microsystems, Inc.

[9] Sybase is a trademark of Sybase Corporation.

the ICN. The Common File System (CFS), our mass file system, stores data from each partition separately. It prevents users logged onto lower-partition machines from accessing files stored in higher-level partitions. The Security Assurance Machine (SAM) authenticates and records all attempts to down-partition files within CFS. In the future we plan to add two other service nodes to the system. The Facility for Operator Control and User Statistics (FOCUS) provides operations control, batch job scheduling, and accounting control for the ICN. The Print and Graphics Express Station (PAGES) provides hardcopies for ICN users.

The content of the audit records produced by the three service nodes currently targeted differs according to the tasks they perform. NSC audit logs contain fields about logons, while CFS and SAM logs contain fields about file handling activities. Differences in auditing software, developed independently for each system, lead to differences beyond these necessary ones. Nevertheless, the three data sources contain similar kinds of information. Each audit record describes a single event, whether an attempted ICN authentication (NSC) or an attempted file activity (CFS and SAM). Both failures and successes are recorded. All audit records contain a unique ID for the ICN user, the date and time of the user's activity, the ICN charge code[10] used, and a code indicating what type of error, if any, occurred. The rest of the record describes the event itself. For the NSC, this part of the record contains:

- The partition from which the authentication attempt originated.
- The ICN address of the machine from which the authentication attempt originated.
- The partition, classification level, and network component (including SAM) that the user wishes to access.

For the CFS, this part of the record contains:

- The machine from which the request originated.
- The classification of the CFS session.
- The size (in bytes), partition, file name, and location within the CFS directory structure of the file being acted upon. If the command requires two locations (e.g., copy), the old and the new names and locations are listed in separate fields.
- The action recorded.

For SAM, this part of the record contains:

- The name and CFS location of the file to be down-partitioned (where applicable).
- The partition to which the file is to be moved (where applicable).
- The action recorded.

When combined in our database, these records provide a near-complete picture of each user's network-level activity. Addition of the audit records for the job control and hardcopy service nodes will complete the picture.

The remainder of this section describes how we have dealt with the data handling issues outlined in section 2.3.

- **Data parsing** has entailed much work in understanding the data sources and in adapting them to Sybase formats. When addressing a new data source our priority is to obtain a format description and a sample listing. We study these, and meet with the managers of each service node, as necessary, until the audit trail content and format are fully under-

[10] An accounting parameter.

stood. At this point appropriate Sybase formats are chosen for each field and a program (in Transact-SQL[11], a superset of SQL and C) is written to bring the raw data into Sybase. In keeping with our modular approach, parsing programs for the different data sources were developed, and still function, independently.

Parsing difficulties have varied among our data sources. NSC audit logs have a standard record-and-field structure. CFS logs originally have an idiosyncratic structure, but are reformatted into a standard structure by CFS personnel for NADIR. Our parsing problems for these data sources have therefore been confined to formatting. For example, file sizes, as generated by the Crays and recorded by the CFS logs, often exceed Sybase's 32-bit integer capacity, forcing them to be formatted as floating point numbers. SAM audit logs have an idiosyncratic structure akin to the original CFS structure. As SAM personnel have not reformatted these logs, this has entailed a significant effort by NADIR personnel to accommodate the logs to a record-and-field structure. Each user session is assigned a unique process ID number that we use to identify and assemble the user's records for that session.

- **Data rationalization** has been a major effort. During our initial study of each service node we can normally identify meaningless fields. We do read these into the database for completeness, though we do not make further use of them when deriving profiles. It is also somewhat straightforward to eliminate defective records. More difficult is the task of rationalizing errors and undesirable audit record features. We have identified these by graphing a representative set of data from each field and for each combination of related fields, checking this output for consistency with the expected state of affairs. Sybase has some simple capabilities that can help in this process, e.g., by tallying the frequency of occurrence of a categorical variable. We have also made use of statistics and graphics software for this purpose. Once error types and undesirable features for a given service node have been identified, all steps required to rationalize them are added to the parsing program.

  Our worst data rationalization problems have concerned the CFS audit records, which are problematic for all the possible reasons outlined in section 2.3. The CFS auditing system is old, has been modified often, and monitors a complex set of tasks. It generates records with respect to CFS actions in response to a user's request, whereas we take the perspective of the user. Thus a single user request to CFS can generate more than one audit record -- or, when some error condition prevents CFS from taking an action, a defective or garbled record.

- **Data fusion.** NADIR has thus far been developed in three stages, first using NSC data, then adding CFS, and later SAM. Data fusion problems have arisen in cases where the content or formatting of fields in later-incorporated service nodes conflicted with standards set in dealing with earlier-incorporated nodes, especially the NSC. An example is the difference between NSC and CFS audit logs in their representation of charge codes, as described earlier in this section. CFS and SAM time and data fields also differed from the NSC format. We have always resolved such problems by adapting the later nodes to the standards developed for the NSC. This seems a natural methodology for multi-source databases.

- **Data aggregation.** All instances of data aggregation in our database involve CFS or SAM data. Error codes for each of these systems are grouped into a few main categories,

[11] Transact-SQL is a trademark of Sybase Corporation.

such as path errors (e.g., named file does not exist) and classification errors. In the CFS field listing the action requested, we aggregate the over two dozen actions defined by CFS into ten categories so that we can see general activity patterns. For example, *save*, *replace*, and *store*, which all save files but differ in how they treat pre-existing files with the same name, are aggregated into one category *put*. In the SAM field showing the action taken, four types of logouts are merged into one.

- **Data reduction** has been a key part of our methodology. While we have only reduced data across fields for data fusion purposes, we have made heavy use of data profiling across records. Profiling serves three purposes. First, it vastly reduces the size of the database, especially for CFS data. CFS usually generates over 30 megabytes of raw audit data a week, which profiling reduces to approximately a quarter of a megabyte. Second, profiling provides an easy-to-understand summary of the data, suitable for examination by hand or graphically. The profiled data are so useful that we rarely examine the raw data except to follow up on anomalies detected in the profiles. Third, the scope of the data reduction gives us the luxury of deriving new fields without overloading the database. For example, usage can be broken down by day of week and time of day. We will describe the profiling process in greater detail in section 3.4.

3.2 Data transmission

As described in section 2.2, data transmission has been a necessary step while integrating service node data into a complete picture of ICN usage. Audit records from the NSC, CFS, and SAM are periodically piped to CFS. An automatic procedure copies them daily onto a DEC-8250[12] and from there into NADIR. NADIR profiles the raw data weekly and saves the profiles in CFS. This process has six advantages:

- Profiles generated from all data sources can be analyzed and reported together.
- The data are available both in raw form, for in-depth analysis, and in summary form (profiles). Raw and profiled data are stored permanently on CFS. The;· can be read back into NADIR for comparison with current data or for performing long-term queries such as background checks on particular users.
- Changes to existing auditing systems and effort by auditing system personnel are minimized.
- NADIR implementation has not measurably degraded the performance of any target system.
- Our SPARCstation currently does not have enough memory capacity to hold a week's worth of CI·S audit logs in ascii form and in Sybase form simultaneously. The DEC-8250 provides an intermediary platform from which the data can be read into Sybase, thus finessing this memory problem.
- Audit logs from .he three service nodes are transmitted separately, in keeping with our modular approach. If logs from one node are unavailable, this does not affect transmission of the others.

3.3 Data security

We always take care to protect the integrity of the audit record, as this is critical to ensure the validity of the intrusion and misuse detection process. ICN audit records have always been treated as sensitive at Los Alamos because of their importance for both security and accounting. Access to them on the target systems and on CFS is restricted to several system

[12] DEC-8250 is a trademarksof Digital Equipment Corporation.

managers, and backups are made routinely. We restrict access to the audit record throughout the process of transmitting it to NADIR and analyzing it. Audit records are kept in a secure partition of the ICN and transmitted via secure lines. Any obvious tampering would be detected by NADIR. For example, a change to data from only one service node would leave tell-tale discrepancies between the nodes. If the tampering took place after the data left CFS, changes also could be found by comparing NADIR's version of the data with a clean copy from CFS. For these reasons we believe that NADIR implementation has enhanced the security of our audit data.

User privacy gives us an additional incentive to protect the audit records, the NADIR database, and especially NADIR reports. We protect user privacy by keeping the results of our analysis confidential. The names of, and details about, suspicious users and events are not discussed outside closed meetings. Printed reports are treated as sensitive and circulated to a limited set of individuals on a need-to-know basis.

3.4 Data profiling

NADIR maintains profiles for each individual ICN user and for a composite of all ICN users. As described earlier, the profiles summarize the raw audit data, making it easier to store, understand, and analyze. At this point in development, new profiles are generated for each week. As each audit record passes from the DEC-8250 onto NADIR, it is parsed into fields and the appropriate counts in the profiles are incremented. At the end of the week the profiles are saved to CFS for permanent storage.

3.4.1 Individual user profiles

Individual user profiles provide a weekly summary of network behavior for each authorized user of the ICN. These profiles can be conceptualized using the same record-and-field model as the raw audit data. Each profile record describes an individual user (as opposed to a user's action, in the raw data). Each profile field describes some characteristic of that user, derived from the raw data. Profile fields are grouped into three sections:

- *User Definition* fields (Table 3.4-1) provide basic information about each ICN user. They are initialized when an individual first becomes an authorized ICN user. The user number (which defines the user) never changes, while information such as name and user group change only as circumstances require.

Table 3.4-1: Individual User Profile: User Definition

User Number	The user's unique ICN identification number.
User Name	The user's name.
User Type	One of various types of ICN users, some with special privileges.
User Group	The group or organization for which the user works.
Mail Stop	The user's Los Alamos mailing address.
Citizenship	The user's citizenship.

- *User History* fields (Table 3.4-2) quantify or list the different types of behavior performed by the user. For example, one quantity field holds the number of distinct charge codes used by the user during the week, and one listing field holds the names of CFS files accessed during the week.[13] These fields can be used to decide which activities are normal for the user.

Table 3.4-2: Individual User Profile: User History

NSC Component	
Sources	The quantity and list of the different source machines from which the user has attempted to log on to the ICN.
Destinations	The quantity and list of the different ICN machines to which the user has attempted to log on.
Charge Codes	The quantity and list of the different charge codes with which the user has attempted to log on.
Blacklists	The number of times, and the last date, upon which a user has been blacklisted; i.e., has had his or her ICN privileges taken away.[14]
SAM Component	
Validation	A single bit, on or off, indicating whether the user was validated on the ICN (as shown by the existence of an NSC profile) during the week in question.
Charge Codes	A list of different charge codes used on SAM.
CFS Component	
Dataset Paths	The quantity and list of data paths (or files) the user has attempted to access.
Sources	The quantity and list of different worker machines from which the user has attempted to access CFS.
Charge Codes	The quantity and list of different charge codes used on CFS.

- *User Activity* fields (Table 3.4-3) hold the actual count statistics for different types of user activity. Examples are numbers of SAM logons or CFS password errors.

The data compression gained by profiling enables us to derive profile fields that are more fine-grained, and therefore more numerous, than the raw data fields. Multiple fields arise in three ways. First, two user history fields (quantity and list) can be generated by a single raw data field. Second, separate user activity fields can address different subcomponents of raw data fields. For example, the frequencies of occurrence of the ten CFS actions, all from a single raw data field, are tallied into separate user activity fields. Third, information from more than one raw data field can be pooled to define more specific user activity fields. Thus frequencies of NSC logons, CFS accesses, and SAM accesses are broken down by time of day and day of week. For most user activities successful and unsuccessful actions are tallied into distinct fields. Because of these factors, the original set of 38 raw data fields yields 147 profile fields for each user.

[13] More accurately, each quantity field is keyed to a separate table which contains the corresponding list for each user. The table containing the lists of SAM charge codes also contains the count statistics for each charge code.

[14] Individuals are blacklisted if they have five sequential authentication failures, or under certain other circumstances of unauthorized behavior. Reinstatement of ICN privileges must be approved on a case-by-case basis by security personnel.

Table 3.4-3: Individual User Profile: User Activity

NSC Component	
Source	Eight counters that tally successful and unsuccessful logons from source machines in each of four partitions.
Destination	Eight counters that tally successful and unsuccessful logons to destination machines in each of four partitions.
Classification	Eight counters that tally successful and unsuccessful logons at four possible computing classification levels.
Time of Day	Six counters that tally successful and unsuccessful logons during one of three shifts (day, swing, and night).
Day of Week	Four counters that tally successful and unsuccessful logons on weekdays versus weekends.
SAM accesses	Two counters that tally successful and unsuccessful attempts to log on to SAM through the NSC.
SAM Component	
Logons	One counter that tallies successful SAM logons.[15]
Movements	Two counters that tally successful versus unsuccessful SAM file movements.
Day of Week	Four counters that tally successful and unsuccessful file movements on weekdays versus weekends.
Time of Day	Six counters that tally successful versus unsuccessful file movements during one of three shifts (day, swing, and night).
Charge codes	Counters that tally attempted file movements using each charge code specified in the SAM component of the user history.
Partitions	Four counters that tally attempted file movements into each of four ICN partitions.
Errors	Seven counters that tally different types of user errors.
CFS Component	
Commands	Two counters that tally all successful and unsuccessful CFS commands.
Command Type	Twenty counters that tally successful and unsuccessful CFS commands, for each of ten command types (save, get, etc.).
Source Partition	Eight counters that tally successful and unsuccessful CFS commands from machines in each of four partitions.
File Partition	Eight counters that tally the number of successful and unsuccessful CFS commands, for files in each of four partitions.
Classification	Eight counters that tally successful and unsuccessful CFS commands in four possible computing classification levels.
Time of Day	Six counters that tally successful and unsuccessful commands during one of three shifts (day, swing, and night).
Day of Week	Four counters that tally successful and unsuccessful commands on weekdays versus weekends.
Errors	Seven counters that tally different types of user errors.

[15] Unsuccessful SAM logons are blocked by the NSC, and therefore are never recorded by the SAM logs.

3.4.2 Composite user profiles

The composite user profile provides a weekly summary of network behavior on the ICN as a whole. Its structure is that of one record with many fields, each of which describe some characteristic of the system derived from the raw data. Most characteristics are derived separately for each hour of the week. The composite user profile is detailed in Table 3.4-4.

Table 3.4-4: Composite User Profile

NSC Component	
Valid Logons	The number of successful logon attempts to the NSC during a given hour.
Invalid Logons	The number of unsuccessful logon attempts to the NSC during a given hour.
Errors	The number of logon errors during a given hour, for each of thirteen error types.
SAM Component	
Charge Codes	The number of different charge codes used on SAM during the week, and the number of accesses made using each code.
Logons	The number of logons to SAM during a given hour.
Users	The total number of different users who logged onto SAM during a given hour.
Activity	The number of successful versus unsuccessful down-partitions during a given hour.
Partition	The total number of successful SAM file movements into each of four partitions during a given hour.
Errors	The total number of SAM errors during a given hour, total and for each of seven error types.
CFS Component	
Valid Commands	The total number of successful CFS commands during a given hour.
Invalid Commands	The total number of unsuccessful CFS commands during a given hour.
Command Type	The total number of successful and unsuccessful CFS commands during a given hour, for each of ten command types.
Source Partition	The total number of successful and unsuccessful CFS commands from machines in each of four partitions during a given hour.
File Partition	The total number of successful and unsuccessful CFS commands affecting files in each of four partitions during a given hour.
Classification	The total number of successful and unsuccessful CFS commands in four possible computing classification levels during a given hour.

3.5 Expert rules

Profiles are checked against expert rules that encode scenarios of security policy violation or suspicious activity. One set of rules pertains to individuals. Before these rules are applied, each user is assigned a Level-of-Interest of zero, indicating that there is no reason to consider him or her suspicious. The individual user profiles are then checked against the rules, with the Level-of-Interest incremented each time a user matches a rule's scenario. The amount of the increment is determined by the rule's importance. After this process, the users with the highest Level-of-Interest are reported for further investigation. A similar process applies to the composite user profiles: scenarios considered suspicious for the ICN as a whole are encoded in rules against which each profile is checked. The resulting Level-of-Interest quantifies the apparent overall integrity of the system.

3.5.1 Rule development

In developing our expert rule set, an important first step was interviewing the experts -- ICN security personnel. Interviews of administrators charged with establishing and enforcing the Laboratory's security policy were somewhat straightforward. The Laboratory's security policy is well defined and documented, and we quickly encoded it into expert rules. Interviewing security auditors was more timestaking but extremely fruitful. We found that auditors rely on an undocumented combination of extensive knowledge of the ICN, experience with previous intrusions or misuses, and instinct. Elucidating and formalizing this knowledge took several weeks and resulted in an expanded set of rules. For SAM, the data had been parsed and analyzed weekly for two years before being moved to NADIR. The rules developed in this process were easily incorporated into our rule set.

Another important part of our rule development was a statistical analysis of the data. From a group of over 4000 individuals and the composite user profiles we calculated the characteristics of average user and system behavior. Then we identified specific individual and composite user profiles that deviated significantly from the norm. Review of these profiles, and the corresponding raw data, enabled us to figure out which of these deviations, particularly if combined with other indications, comprised a suspicious event. Dynamic graphical data analysis, which permits active analyst intervention, was also applied to the user profiles. It was ideal for spotting extraordinary usage profiles within a large data structure such as the individual user profiles.

This process of interviews and statistical analysis led to the definition of an initial rule set, which we then tested repeatedly against the profiles. This testing phase led to the discovery of misuse scenarios that had not previously been identified, and the implementation of new rules to detect them. Often these scenarios and rules were much more elaborate than those used in traditional manual audit reviews. We also discovered previously undetected system vulnerabilities. Where these could not be remedied, we added rules to monitor them closely. This process of testing and revising our rule set is an ongoing one, as we continually aim to improve the accuracy of our system.

3.5.2 Rule content

Most of our expert rules are geared to the detection of the four security violation types outlined in the introduction: disclosure of information, violation of integrity, denial of service, and masquerading. While our exact rules are classified, we present below some generic types of behavior we look for on the different service nodes.

Disclosure of information can be signalled by suspicious file activity. A perpetrator intending to disclose information may browse through the file system in search of worthwhile material. This can be evidenced on CFS by an unusual number of file accesses, especially reads and copies, for an individual or a composite user profile. It also can cause many file access errors as the browser attempts to read or copy files for which he or she is unauthorized. On SAM, attempted disclosure of classified information can be evidenced by many file movements into the open partition, or many errors when attempting to perform such file movements. Such activity is especially suspicious for specific users, groups of users, and source partitions.

Violation of integrity can be signalled by other kinds of suspicious file activity. Specifically, it can be evidenced on CFS by an unusually high number of file accesses, especially deletes and modifies, and many errors when attempting these access types. Violation of integrity also can spawn errors by innocent users when they attempt to access files that a perpetrator has deleted.

Denial of service can be detected primarily by signs of system overload. A perpetrator can engineer massive numbers of logon attempts (NSC), file access attempts (CFS), or file movement attempts (SAM), thus hanging up the relevant service node or nodes. This would be evidenced by an unusually high number of action attempts and action failures. Massive file deletions on CFS could signal an attempt to deny service by making files unavailable.

Masquerading can be indirectly indicated by signs of disclosure or violation of integrity, as these are likely goals for a successful masquerader. In particular, a masquerader is likely to move files within or between partitions, or attempt to modify file permissions so that he can more easily access them in the future. He is also likely to browse through the system, as described above. The act of masquerading itself also can leave noticeable traces. On the NSC, masquerading attempts can result in an unusually high number of logon attempts, and an unusually high ratio of logon failures to logon attempts, both for specific individuals whose accounts are under attack and for the network as a whole. Composite NCS profiles also can contain many 'user unknown' errors. Individual or composite profiles can show a discrepancy on all service nodes between normal working hours and the off-hours, when many attacks occur.

NADIR also includes rules encoding specific security policies. For example, users are not supposed to write automated logon procedures, since these involve writing one's password into a file. We have rules to flag behaviors that indicate the use of these procedures, such as a sequence of repeated invalid logon attempts involving an identical error.

3.5.3 Rule implementation

Our expert rules are encoded in a condition-action (if-then) form. The condition ('if') is a profile scenario considered to be suspicious or a violation of security policy. The action ('then') is a specified incrementing of the Level-of-Interest of the relevant user (or composite user profile). As the rules are applied to the profiles, a matrix called the Anomaly Record is built that lists the rules triggered, and the Level-of-Interest, for each user and for the system as a whole. An example of a rule that applies to an individual user profile is the following, which focuses on logon failures:

> **IF** the Failure Ratio of a user is >n1[16],
>
> **AND** the user has logged on >n2 and ≤n3 times,
>
> **THEN** update the Anomaly Record, and increment the user's Level-of-Interest by n4.
>
> **EXPLANATION:** In this rule, Failure Ratio is the ratio between the user's unsuccessful and (total) attempted logons. If a user has logged onto the ICN at least n2 times then the user is not new to the ICN. A Failure Ratio of n1 is significant because it greatly exceeds that of the average ICN user. The value of n4 depends on the user's Failure Ratio and his or her total number of logons.

The following is an example of a rule applying to a composite user profile:

[16] As our specific rules are sensitive, we will use variables (n1, n2, etc.) in place of the actual numbers.

IF "Unknown User" errors are >n1 per hour,
OR >n2 per day,
OR >n3 per week,
THEN update the Anomaly Record, and increment the system's Level-of-Interest by n4.
EXPLANATION: This rule exploits the fact that the rate of logon failures due to an invalid user number is statistically very consistent. Extreme variations from this expected activity could be a sign of a break-in attempt. The value of n4 here depends on the specific frequency of these errors.

3.6 Reporting

NADIR currently generates weekly hardcopy reports. For each service node there is a one-page summary of all activity, e.g., the number of users and the number of successful and unsuccessful user requests. For each node there is also a set of graphs of different types of user activity, plotted over time with a granularity of one hour. These are useful for visually spotting abnormal patterns, e.g., an unusual spurt of off-hour usage. The rest of the report summarizes the results of the expert rule analysis. Suspicious users are listed in descending order of their Level-of-Interest, with a list of the rules each has triggered. The Level-of-Interest of the system, and a list of its triggered rules, also are reported.

Security auditors review each week's report. They examine each anomalous user's behavior and decide whether to investigate by further analyzing the user's audit data or interviewing him. An investigation may result in a warning to the user, or the user's being blacklisted from the system. More often, it results in a learning experience for the user, who is helped to correct technical, and sometimes security, errors. The auditors file a short report at the completion of each investigation, giving details of its resolution. These reports, and periodic reviews of NADIR by the security auditors, provide valuable feedback from which we continually try to improve the system.

While these weekly printouts are the main means by which auditors access NADIR output, they are not the only means. NADIR also produces a more detailed weekly report that includes data from the audit record. This report is stored in the Secure partition of the ICN's Common File System, where it may be accessed and reviewed electronically by authorized personnel. Besides weekly reporting, NADIR can produce reports on demand. On-the-spot reports have proved invaluable in analyzing ongoing emergencies detected by system operators. We plan to add an alarm component to the system so that NADIR itself can alert operators to such emergencies. Finally, raw or profiled data stored on CFS also can be used on an ad-hoc basis to perform background analyses of current and past activity for a particular user or users. Users can examine data using Sybase's built-in facilities, or pipe data to a statistical software package for more detailed analysis.

4 Results

The difficulty of evaluating intrusion detection systems is well documented as a chronic problem in the field [2, 11]. To quantify a system's success one needs to know its frequencies of false positives (innocent actions detected as anomalies) and false negatives (failures to detect actual intrusion or misuse attempts). One also must decide what levels of false positives and negatives are tolerable for the system to be considered successful.

Our false positive rate is high, but within tolerance given the interactive usage for which the system is designed. We have detected many 'true positives', including:

- automated logons,

- misuse of special-use user numbers,

- attempted (unsuccessful) logons using another person's user number,

- attempted (unsuccessful) logons from terminals in partitions to which the user did not have access,

- and attempted (unsuccessful) logons to computers in partitions to which the user did not have access.

However, most of the flagged individuals and events have not been intruders, spies, or even users deliberately misusing the system. Nevertheless, we designed the system to *assist* the hands-on auditing process and to be used as an educational tool. Therefore, we regard this high false positive rate as allowable if the list of flagged users and events is short enough for quick review.

False negatives are much more difficult to quantify than false positives, lacking independent means of identifying intrusions and misuses. We can only report that in over two years of active NADIR usage we have yet to discover by other means any misuse or intrusion attempt within NADIR's scope which NADIR did not also detect. We have also successfully detected ICN intrusion attempts staged by security officers.

From a technical perspective, the successful implementation of NADIR demonstrates the feasibility of extending automatic security auditing to the network level. Our design choices, especially the system's modularity, its profiling capacity, and its flexible rule base, have resulted in a system that is simple and easy to modify. Serendipitously, NADIR has also turned out to be a versatile network management tool. It has enabled us to detect hardware and software problems, and security vulnerabilities, in many parts of our network. Moreover, its detailed statistics on network activity have proved useful in such areas as accounting and network planning.

5 Future directions

Our main goal is to enable near real-time intrusion and misuse detection. Toward this end we have already implemented a smaller NADIR clone directly attached to the NSC. The clone receives NSC audit records within ten minutes of the event recorded, and will soon be modified to receive them within several seconds. As on the main NADIR system, profiles can be generated on demand; this has already proved extremely valuable in assessing ongoing system abnormalities. We also compare clone-derived profiles with those derived by our main system as a double-check on the main system. We plan to implement similar clones on the other service nodes as well, replacing our current system with a network of workstations, each contributing to a distributed database. Once these clones are in place we also hope to implement rules that analyze individual audit records as they are piped to the clones, and an alarm system that notifies system managers of particularly suspicious ongoing activity.

Other future priorities are:

- To complete the process of adding and refining expert rules for CFS and SAM.

- To incorporate data from the other ICN service nodes into the system.

- To supplement our expert rulebase with a true anomaly detection component that 'learns' typical behavior for each user, then reports deviations from these norms.

We also are interested in applying our expertise to large databases beyond the computer security arena. We believe that audit logs from other complex systems, such as credit card transactions, medical care payments, and procurement systems, can be analyzed using NADIR-like user and system profiles and expert rules. For example, rules could flag abnormal numbers of credit card transactions by a particular user or medical care payments to a particular doctor, just as NADIR rules flag users with unusual numbers of logons.

It is the prospect of broad applications that makes the idea of general tools for the analysis of large databases so attractive to us. Currently, the analyst must write new software for parsing, rationalizing, and profiling the audit record in each application. He or she also must write the expert rules and the reporting programs 'from scratch'. This state of affairs forces the analyst to duplicate the effort already undertaken on other projects. If tools existed to guide the analyst through the task of creating a new application, giving him or her the necessary freedom to select appropriate field formats, profiling fields, etc., development time could be vastly reduced. At Los Alamos, a working group on anomaly detection is tasked to define common needs for such tools and seek funding to build them. We welcome readers' input in this effort.

Acknowledgements

We acknowledge the contributions of Alex Marusak, Sharon Wilhemy, Wanda Dunlop, and Lori Kelley, who review the weekly NADIR reports and who have been a valuable source of feedback on NADIR's functioning. We thank the many service node personnel who cooperated with us, and the members of the Operational Security Division who contributed to our rule base.

References

[1] C. Browne. *DOTTYE - An Interlisp Program for the Analysis of the Network Security Controller Files* (Los Alamos National Laboratory Technical report, January 1984).

[2] J. Campbell, B. Miller, P. Proctor. *Requirements Definition and Computer Misuse Detection Systems Analysis (Final)* (SAIC, Comsystems Division, February 1991)

[3] D. Denning and P. Neumann. *Requirements and Model for IDES - A Real-Time Intrusion Detection Expert System, Final Report* (Computer Science Laboratory, SRI International, August 1985).

[4] D. Denning. *An Intrusion Detection Model* (IEEE Transactions on Software Engineering, Vol. 13, No. 2, February 1987).

[5] D. Denning, D. Edwards, R. Jagannathan, T. Lunt, P. Neumann. *A Prototype IDES: A Real-Time Intrusion Detection Expert System* (Computer Science Laboratory, SRI International, August 1987).

[6] L. Halme and B. Kahn. *Building a Security Monitor with Adaptive User Work Profiles* (Proceedings of the 11th National Computer Security Conference, October 1988).

[7] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber. *A Network Security Monitor* (Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1990).

[8] L. Heberlein, K. Levitt, and B. Mukherjee. *A Method to Detect Intrusive Activity in a Networked Environment* (Proceedings of the 14th National Computer Security Conference, October 1991)

[9] L. Heberlein, B. Mukherjee, and K. Levitt. *Internetwork Security Monitor: An Intrusion Detection System for Large-Scale Networks* (Proceedings of the 15th National Computer Security Conference, October 1992)

[10] J. Hochberg, K. Jackson, J. F. McClary. *The Insider Threat to Computerized Information* (United States Department of Energy, Office of Safeguards and Security, July 1992, LA-UR-92-2258)

[11] B. Hubbard, T. Haley, N. McAuliffe, L. Schaefer, N. Kelem, D. Wolcott, R. Feiertag, M. Schaefer. *Computer System Intrusion Detection* (Trusted Information Systems, Inc., September 1990, TIS Report #348)

[12] K. Jackson, D. DuBois, and C. Stallings. *A Phased Approach to Network Intrusion Detection* (Proceedings of the United States Department of Energy Computer Security Group Conference, May 1991, LA-UR-91-334).

[13] K. Jackson, D. DuBois, and C. Stallings. *An Expert System Application for Network Intrusion Detection* (Proceedings of the 14th National Computer Security Conference, October 1991, LA-UR-91-558).

[14] G. Liepins and H. Vaccaro. *Intrusion Detection: Its Role and Validation* (Computers & Security, Elsevier Science Publishers Ltd, August 1992)

[15] T. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. Edwards, P. Neumann, H. Javitz, A. Valdes. *IDES: The Enhanced Prototype A Real-Time Intrusion Detection Expert System* (SRI International, October 1988).

[16] T. Lunt. *Real-Time Intrusion Detection* (Proceedings of COMPCON, Spring 1989).

[17] T. Lunt, R. Jagannathan, R. Lee, A. Whitehurst. *Knowledge-Based Intrusion Detection* (Proceedings of the 1989 AI Systems in Government Conference, March 1989).

[18] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, C. Jalali. *IDES: A Progress Report* (Proceedings of the 6th Annual Computer Security Applications Conference, December 1990).

[19] M. Sebring, E. Shellhouse, M. Hanna, R. Whitehurst. *Expert Systems in Intrusion Detection: A Case Study* (Proceedings of the 11th National Computer Security Conference, October 1988).

[20] S. Snapp, J. Brentano, G. Dias, T. Goan, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance, D. Teal, and D. Mansur. *DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and Early Prototype* (Proceedings of the 14th National Computer Security Conference, October 1991)

[21] C. Stoll. *The Cuckoo's Egg* (New York, Doubleday, 1989)

[22] G. Tsudik and R. Summers. *AudES - An Expert System for Security Auditing* (Proceedings of AAAI Conference on Innovative Applications in AI, May 1990).

[23] J. Winkler. *A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks* (Proceedings of the 13th National Computer Security Conference, October 1990).

[24] J. Winkler. *Intrusion and Anomaly Detection: ISOA Update* (Proceedings of the 15th National Computer Security Conference, October 1992)