

Conf-9410212--2

LA-UR 94-3385

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: AN AUTOMATED COMPUTER MISUSE DETECTION SYSTEM FOR UNICOS

AUTHOR(S): Kathleen A. Jackson, Michael C. Neuman, Dennis D. Simmonds, Cathy A. Stallings, Joseph L. Thompson, and Gary G. Christoph

SUBMITTED TO: 1994 Cray Users Group (CUG) Conference
October 10-14, 1994
Tours, France

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.

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.

MASTER

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

AN AUTOMATED COMPUTER MISUSE DETECTION SYSTEM FOR UNICOS*

Kathleen A. Jackson, Michael C. Neuman, Dennis D. Simmonds,
Cathy A. Stallings, Joseph L. Thompson, and Gary G. Christoph

Computing, Information and Communications Division
Los Alamos National Laboratory
Los Alamos, New Mexico U. S. A.

Abstract

An effective method for detecting computer misuse is the automatic monitoring and analysis of on-line user activity. This activity is reflected in the system audit record, in the system vulnerability posture, and in other evidence found through active testing of the system. During the last several years we have implemented an automatic misuse detection system at Los Alamos. This is the Network Anomaly Detection and Intrusion Reporter (NADIR). We are currently expanding NADIR to include processing of the Cray UNICOS operating system. This new component is called the UNICOS Realtime NADIR, or UNICORN. UNICORN summarizes user activity and system configuration in statistical profiles. It compares these profiles to expert rules that define security policy and improper or suspicious behavior. It reports suspicious behavior to security auditors and provides tools to aid in follow-up investigations. The first phase of UNICORN development is nearing completion, and will be operational in late 1994.

1. Introduction

The goal of computer misuse and intrusion detection is to discover security violations on computer systems. Perpetrators may be either insiders (authorized users) or outsiders who clandestinely access a system. The first line of defense against all such violations is the institution of formality of operations. This is a way of doing business that emphasizes safeguards and accountability. Formality of operations includes institutional practices such as training, configuration management, and physical security measures.

However, several factors limit the efficacy of these measures. The first is human nature. Users often see security as an unwelcome diversion from the main thrust of their work. They resist learning security measures and procedures and frequently fail to apply them. Second, system managers must effect a compromise between conflicting concerns. For example, while it is more secure to compartmentalize activity, today's users require access to distributed resources. Third, systems frequently contain undetected vulnerabilities to attack and misuse. Finally, there is the threat of insiders who deliberately misuse their legitimate privileges [7].

Given these weaknesses, a second line of defense against abuse is the maintenance and analysis of system audit records. In theory, one can detect break-in attempts and other security violations by detecting abnormal or invalid user activity, changes in the system vulnerability posture, and other misuse indications. However, the traditional approach of manual analysis has generally proved unworkable. Human limitations restrict manual review to a sampling or cursory scanning of the large quantity of audit data and system status information typically gener-

*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.

ated. This approach can target only a few obvious misuse scenarios; it may miss even these because of human error and because of the speed at which computer misuse occurs.

The limitations of manual review have long been apparent to security personnel at Los Alamos. While manual review by security auditors did reveal instances of misuse, there was no way to evaluate the general success or completeness of this effort. Large-scale manual audits of past data also proved cumbersome and time-consuming. It was obvious that automated review would be more effective. Such an analysis combines two essential components. First is the expert's knowledge of security problems. Second is the computer's ability to process and correlate, rapidly and accurately, large quantities of data. In addition, the speed of machine processing can allow an automated system to inform auditors of suspicious activity in time for them to trace and stop it. A system can even be programmed to undertake defensive measures itself, such as logging out a suspected intruder or removing a vulnerable machine from a network.

Los Alamos began developing a misuse detection system in the late 1980s; it has been operational since 1990 [2, 5, 6]. This system, called the Network Anomaly Detection and Intrusion Reporter (NADIR), evaluates the security of our main computing network, the Integrated Computing Network (ICN). NADIR monitors several ICN nodes that handle network services such as authentication, and file access, movement, and long-term storage. It analyzes the audit records kept by these nodes, checking for a set of suspicious activities. It uses an *expert system* methodology in that its misuse scenarios were derived from interviews with security experts, and from hands-on examinations of audit record data.¹ NADIR currently does not analyze audit data from ICN host computers (i.e., computers to which normal users have controlling access), such as our six Cray supercomputers.

This paper describes our recent effort to add a new NADIR component that addresses the security of the Los Alamos Crays themselves. Since the Crays run the UNICOS operating system, this component is called the UNICOS Realtime NADIR, or UNICORN. UNICORN has a distributed design: a Cray-based "client" collects data in near realtime and transmits it to a workstation-based "server" for processing. The server analyzes the data, generates reports, and notifies appropriate personnel.

UNICORN differs significantly from the other components of NADIR in that it monitors host computers rather than network service nodes. It differs significantly from other misuse detection systems with which we are familiar, in that it combines two distinct computer security techniques. UNICORN looks both for suspicious *behaviors* and for suspicious *characteristics*. In the first category, it analyzes system audit records for evidence of suspicious behavior (as does NADIR). In the second category, UNICORN analyzes the status the Crays for characteristics that indicate a vulnerable configuration or other evidence that misuse has taken (or is taking) place.

In developing UNICORN we maintained the design philosophy that served us well with the original NADIR [8]. UNICORN is modular. It both integrates and separates information within different modules so that we will be able to easily analyze data from several Crays simultaneously. It will enable us to take individual target Crays in or out of the analysis system (either deliberately or because of failures). It checks all data fed into the database for errors and reduces it to summary profiles. It is designed to undertake data collection and analysis while avoiding any disruption of the normal conduct of business.

¹This methodology is in contrast to *anomaly detection* systems, which compile statistics of normal behavior, then note deviations from these norms.

UNICORN is nearing the end of Phase 1 development. In this phase we focused on writing client and server software that performs all basic required functions on a relatively small set of data. For development purposes, we have installed client software on a single Cray and server software on a single workstation. Section 2 describes the UNICORN system to date. We expect to spend the next year in further development. This will include expanding the client and server software to analyze a more comprehensive data set, and installing client software on additional Crays. Section 3 describes our status, and outlines our plans for the future, in more detail.

2. System Description

UNICORN is a distributed system. A Cray-based "client" collects and transmits data. A workstation-based "server" confirms the integrity of the received data, formats it into a canonical form, profiles the data, analyzes it for signs of misuse, and produces reports or alarms as required. Manual review of suspicious events takes place off-line. Every day we back up the transmitted data, the profiled data, and all reports to permanent file storage. Figure 2-1 illustrates this process. The following sections detail each of the above activities.

Isolation of the processing and alarm functions in the server has two major advantages. First, it provides a greater level of trust in the detection system. Second, it provides the capability of correlating activity from several Crays, thus increasing UNICORN's sensitivity to misuse distributed over the network. UNICORN could operate entirely on each Cray, however the substantial increase in trust and flexibility that derives from separating the functions easily justifies the cost of a workstation and development of data transmission software.

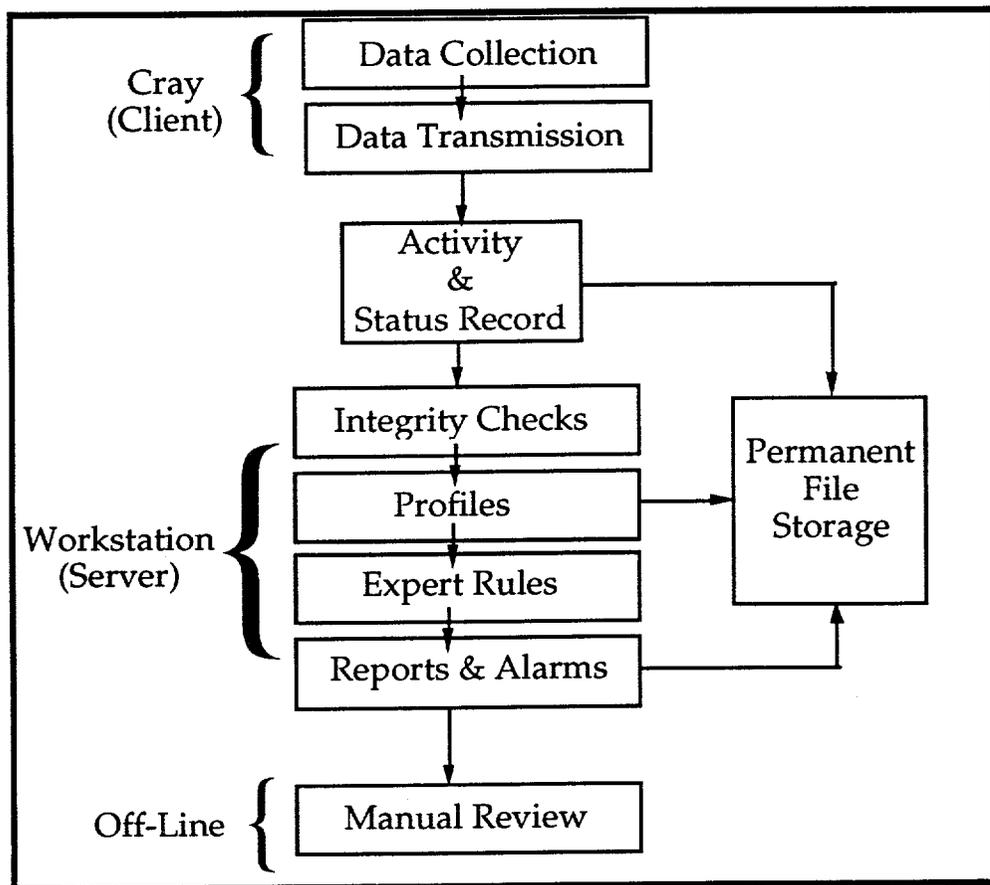


Figure 2-1: UNICORN Distributed Implementation

2.1 Client

One UNICORN objective is to analyze UNICOS activity promptly, that is, in a near realtime² mode. We expect that this will permit quick response to serious events. To meet this objective, each monitored UNICOS Cray executes a client process that collects information and transmits it immediately to a workstation-based server for database insertion and analysis. The client is programmed in the C language. This necessary component of UNICORN performs three functions: it

1. collects selected UNICOS audit logs,
2. probes for signs of misuse and for configuration vulnerabilities, and
3. transmits audit, misuse and vulnerability data to the workstation-based server.

2.1.1 UNICOS Audit Logs

The client currently collects data from the UNICOS security log (SLOG), as defined in UNICOS version 8.0. We plan to expand it to additional logs in the future. The client runs periodically³, saving its current log position for the next collection pass. This approach permits asynchronous retrieval and transmittal of log data from the Cray after system or network interruptions. With each pass, the client parses the audit logs and filters out bad data, preparing it for transmission to the server.

2.1.2 Misuse Characteristics

The client includes an automated security scanner that runs at regularly scheduled intervals⁴. It looks for suspicious *characteristics* on the Cray (as opposed to suspicious *behaviors* noted in the system logs). In this way UNICORN can identify misuse that may not show up in the standard audit record. These include problems with system configuration, and signs of indirect user modification of the system. This component of UNICORN is similar in function to Purdue University's COPS [4] and Lawrence Livermore National Laboratory's SPI [3], though it looks for a wider range of characteristics. It includes enhanced Kuang (expert system) checking [1] for critical activity combinations. The security scanner checks for danger signs such as the following:

- Files modified by a daemon (e.g., sendmail or crontab writing to a file or changing file permissions).
- Minor changes in file permissions with indirect consequences (e.g., a user in a "system" group accidentally makes his or her home directory world writable).
- Valid file accesses with modifications that violate site policy (e.g., root modification of /etc/hosts.equiv, adding a '+').

²At this time, we define near realtime as the detection and reporting of misuse within, at most, one hour of its occurrence.

³Currently, every six seconds. This collection rate is tunable.

⁴The periodicity is variable. For example, every minute the scanner sends all changed udb entries to the server, and every hour it sends the udb state (summary statistics about the udb). Then, as a sanity check, it sends the entire udb to the server once a day.

- Significant changes in the /etc/udb file (e.g., security attribute additions, deletions, or modifications).
- Modification of critical system binaries.
- Flaws in critical file formatting (e.g., the /ect/group and /ect/passwd files).
- The protection status of system directories, files, and devices (e.g., world writable system directories and files, and world readable memory devices).
- Correct anonymous FTP configuration.
- File access permission problems (e.g., world writable files referenced by system crontab entries, world writable files referenced by /etc/rc, the proper configuration of trust files, world writable user critical files (e.g. .rhosts, .login, .cshrc), and world readable .netrc files).
- Insecure daemons (i.e., sensitive programs such as TFTP and REXD).
- Invalid root configuration (e.g., system files and root login files owned by a user other than root, root's umask set incorrectly, hosts.equiv and ftpusers configured incorrectly).
- World writable home directories.

2.1.3 Data Protection

The many users on each Cray pose a serious potential threat to the integrity of UNICOS audit and security scanner data, and the client process. However, we believe the security features of the UNICOS operating system, if properly implemented, are sufficient to protect the data and process from tampering. Several protections, in particular privileged role separation, provide excellent assurance against alteration of security audit logs. Other audit logs may be protected using UNICOS MLS features, such as Mandatory Access Controls (MAC) or Privilege Access Lists (PALs). The client software is protected by UNICOS security features such as privileged role separation. Finally, the client cooperates with the server in performing several integrity checks on all transmitted data (Section 2.1.4).

2.1.4 Data Transmission

The client transmits all data to the server in binary format. Client and server ensure data integrity cooperatively, using the following means:

- A *sequence number* that ensures the detection of repeated, missing, or out-of-sequence data packets. This helps detect not only data that has been deliberately tampered with, but also transmission mistakes resulting from system or client failures. The server logs all sequence failures. Such failures themselves could trigger an alarm.
- A *shared secret* that is used to verify the authenticity of each received packet. The shared secret sent by the client must match that kept by the server. The shared secret consists of a 32-bit key that can be changed as often as deemed necessary by UNICORN. The server discards packets lacking the correct key, and logs all shared key failures.
- A *source identity check* that verifies the source (Cray) identity of incoming data packets, and whether each packet's internal labeling matches that particular Cray machine. The server logs all invalid sources.

Transmitted data is not encrypted because we consider the network segment between the target Cray and the UNICORN workstation both physically and logically secure. The only nodes (machines) allowed on this segment are special-purpose network services that are physically and internally secure. Access to them is limited to authorized network personnel. No computers accessible to normal users are allowed on this segment. Consequently, there is no way the shared secret can be monitored by unauthorized users while transiting this network segment. However, if the need arises, encryption can be easily implemented.

2.2 Server

The server resides on a Tatung SuperComp™ workstation (a SUN™ clone) with 97 MBytes⁵ of memory and two 1.05 GByte disks. The Sybase™ relational database management system is used to organize the data structure and to enable easy data access. The server software is written in the C language and Transact-SQL (Sybase's version of SQL). The server performs five functions: it

1. decodes the incoming binary data from the client and performs integrity checks on that data,
2. formats the data for use by the server,
3. summarizes this "raw" data into profiles of both individual user activity and composite system activity,
4. examines the profiles for signs of misuse, and
5. reports its findings.

2.2.1 Data Receipt

The server decodes each incoming data packet and checks its integrity (Section 2.1.3). It reports any out-of-sequence or apparently bogus (failed the shared secret test) data. It discards duplicated or bogus data packets. It determines the type of data, and activates appropriate routines for parsing and resolution.

2.2.2 Data Formatting

After the server receives a UNICOS audit record, it first parses the record and places it in a canonical format. We do this to provide a standard data interface to the server. This is useful for two reasons. First, we plan to expand the server to process multiple UNICOS audit logs (in the short term) and to other UNIX operating systems (in the long term). With this approach, the server parsing function will not have to be modified to handle different data formats. All modifications will be limited to this one function; the core of the server will remain unchanged. Second, we wanted to implement the standard audit data interchange format currently proposed in the computer security community [10]. Widespread use of this format will allow the sharing of audit record information from different misuse detection systems. Such a common format is much desired by developers of audit record analysis tools. It includes 'wild card' fields that can be used for system-specific information, such as our 'Partition' and 'Compartment' fields, and event-specific information. Each canonical audit record describes a single event, and is formatted as summarized in Table 2-1.

⁵The server currently requires only 32-48 MBytes of memory.

Table 2-1: UNICORN Audit Record	
BASIC DATA	
Timestamp	The date and time at which the activity occurred.
Event Type	The type of event described in this audit record.
Process ID	The current process identifier.
Outcome	The event outcome. If successful, a return code indicates the type of activity. If unsuccessful, an error code indicates the type of failure.
User IDs	A full description of the subject's user identifiers.
Group IDs	A full description of the subject's group identifiers.
Session ID	The session to which the process belongs.
Security Level	The security level of the event subject, whether user or process.
Object Description	Information about the objects affected by the event, if any.
MISCELLANEOUS DATA	
Site-Specific Data:	
Host	The host Cray on which the attempted activity occurred.
Partition	The security partition in which the attempted activity occurred (a Los Alamos specific attribute).
Event Source	The source of the activity. For example, the workstation from which a user logged on.
Compartment	The security compartment of the attempted activity.
Category	The integrity category of the attempted activity.
Event Data:	
Activity Data	The data specific to the type of activity being reported; it describes the event itself. Each Event Type has its own set of possible Activity Data values.

2.2.3 Profiles

The server maintains profiles for each assigned user identifier (UID) and for a composite of all UIDs on the Cray being monitored. The profiles summarize the raw audit data, making it easier to store, interpret, and analyze. Profiles are saved daily to the ICN's permanent file storage. The profiles described in this Section are Phase 1 profiles, and are geared toward examining logon activity, configuration, and misuse data. As we expand the scope of examined activities we will expand the profiles accordingly.

2.2.3.1 Profile Design

Profiles are summary statistics of activity over some defined interval. The server maintains two kinds of profiles; individual and composite. Individual profiles summarize the activity

attributed to specific UIDs. Composite profiles summarize the activity of an entire system. Individual and composite profiles are structured as in Tables 2-2 and 2-3.

Each profile consists of a number of *segments*. Each segment corresponds to a certain time interval. The composite profiles are more detailed than the individual profiles: each full day's data is broken into 24 segments, one per hour. The choice of one hour for the profiles' finest granularity seemed appropriate to us, but is configurable to a shorter or longer period. Each segment has numerous *fields* that summarize some aspect of the subject of the profile (individual user or system) during that time interval. These fields are described in Section 2.2.3.2 and 2.2.3.3. Many of these are count statistics such as the number of logon failures during the interval. These statistics are updated each time a relevant audit record is received.

The first two segments of both profiles describe the *current hour* and *current day* thus far. The remaining segments describe a *moving week* of data, of which the seventh day is the most recent day for which complete data are available. For example, if today is Thursday (the current day), the moving week includes data from the previous Thursday through yesterday (Wednesday). As each current hour is completed the current day segment is updated and the current hour segment is re-initialized. As each current day is completed the current moving week is updated and the current day segment is re-initialized. For example, at the end of Thursday, the moving week shifts to last Friday through Thursday.

segment	interval	
1	current hour	
2	current day	
3	m o v i n g w e e k	day 1
4		day 2
5		day 3
6		day 4
7		day 5
8		day 6
9		day 7

segment	interval	
1	current hour	
2	current day	
3-26	m o v i n g w e e k	day 1 (24 hours)
27-50		day 2 (24 hours)
51-74		day 3 (24 hours)
75-98		day 4 (24 hours)
99-122		day 5 (24 hours)
123-146		day 6 (24 hours)
147-170		day 7 (24 hours)

2.2.3.2 Individual Profiles

Individual profiles provide a summary of activity for each authorized UID on the system. They consist of one record for each unique UID. We group the individual profile fields into three sections:

- *User Definition* fields (Table 2-4) provide basic information and mandatory access control data for each UID. UNICORN initializes these fields for each UID when it is authorized on the monitored Cray. The information for this definition is obtained from the UNICOS

password file (/etc/passwd) and user database (/etc/udb). The UID never changes; other information changes only as circumstances require.

Table 2-4: Individual UID Profile: Definition	
User ID	The baseline user identifier.
Group ID	The baseline group identifier (GID).
User Name	The full given name of the user or process associated with the UID.
User Moniker	Nickname associated with the user.
User Number	The user's unique Los Alamos identification number.
User Type	The types of ICN users, some with special privileges.
Comment	Optional description of the user ID.
Initial Directory	The path to the UID's logon directory.
Shell	The location of the UID's default shell.
Security Compartments	The user's assigned active (or default) and authorized compartments.
Security Levels	The user's assigned maximum and minimum security level.
Integrity Categories	The user's assigned authorized categories.
Integrity Classes	The user's assigned maximum and minimum integrity class.

- *User History* fields quantify different types of selected behavior associated with the UID, and are linked to tables listing these types. For example, one history field holds the number of source workstations used by the UID, and is linked to a table listing the actual workstations. Table 2-5 illustrates the type of data maintained in the UID history.

Table 2-5: Individual UID Profile: History	
Logon Component:	
Partitions	The number and a list of the different security partitions from which the UID has attempted to log on to the Cray, both successfully and unsuccessfully.
Workstations	The number and a list of the different workstations from which the UID has attempted to log on to the Cray, both successfully and unsuccessfully.
Logon User IDs	The number and a list of the different UIDs associated with the primary UID, both successfully and unsuccessfully.
Logon Group IDs	The number and a list of the different GIDs associated with the primary UID, both successfully and unsuccessfully.

- *User Activity* fields hold the count statistics for different types of UID activity. These are derived both from the audit record, and from the active security scanner. The misuse recorded here is that which can be attributed to a specific UID. Table 2-6 illustrates the types of data maintained in the UID profile.

Table 2-6: Individual UID Profile: Activity	
Logon Component:	
Successful logons	Counter that tallies all successful logons.
Unsuccessful logons	Counter that tallies all attempted unsuccessful logons.
Successful logon levels	Counters that tally all attempted logons at four security levels.
Unsuccessful logon levels	Counters that tally all attempted logons at four security levels.
Logon errors	Counters that tally various types of logon failures.
Misuse Indication Component:	
UDB Changes	Counters that tally additions, deletions, and modifications to the UID's record in /ect/udb.

2.2.3.3 Composite Profiles

The composite profile provides a summary of UID activity, misuse indications not attributable to a single or specific UID, and vulnerability posture for the whole Cray. The profile consists of one record for each monitored Cray. Tables 2-7 through 2-9 illustrate the composite profile.

Table 2-7: Composite UID Profile: Activity	
Logon Component:	
Successful logons	Counter that tallies all successful logons.
Unsuccessful logons	Counter that tallies all unsuccessful logons.
Successful logon levels	Counters that tally all successful logons at four security levels.
Unsuccessful logon levels	Counters that tally all unsuccessful logons at four security levels.
Logon errors	Counters that tally various types of logon failures.

Table 2-8: Composite UID Profile: System Vulnerability	
Root errors	Counters that tally the occurrences of system files owned by a user other than root, root umask set incorrectly, and other root configuration errors.
Access errors	Counters that tally the occurrences of sensitive directories, files, and devices that are world writable/readable.
Group file errors	Counters that tally formatting and content errors in /etc/group.
Password file errors	Counters that tally formatting and content errors in /ect/passwd.
User file errors	Counters that tally world writable user critical files (e.g., .rhosts, .login, .cshrc) and world readable .netrc files.
Anonymous FTP errors	Counters that tally incorrect anonymous FTP configurations.
Permission errors	Counters that tally the occurrences of various combinations of permission problems.

Table 2-9: Composite UID Profile: Misuse Indication	
UDB Changes	Counters that tally additions, deletions, and modifications to /ect/udb.
System file changes	Counters that tally all changes to various system files, e.g., telnetd, /bin/login.
Sequence failures	Count of the number of out-of-sequence data packets received from the client.
Invalid keys	Count of the number of invalid data packets received from the client (with an incorrect shared secret).
Invalid source	Count of the number of data packets received from a source other than the target Cray
Invalid label	Count of the number of data packets whose internal labeling is incorrect.

2.2.4 Profile Analysis

The UNICORN server compares the profiles to expert rules that encode our security policy and unusual or suspicious activity. One set of rules applies to individual UID activity, another to composite activity.

2.2.4.1 Evaluation Schedule

Profiles are evaluated using the expert rules described in Section 2.2.4.3. The accumulated hour, day, and week profiles are evaluated separately, using different sets of rules. Evaluation is data driven; the timestamp within the incoming data is used to decide when it is time to evaluate. Evaluation is performed as follows.

At the beginning of a new hour:

1. The hour just finished is evaluated.
2. The hour's data is added to the current day.
3. The day thus far is evaluated.

At the beginning of a new day:

1. The day just finished is evaluated.
2. The oldest day is dropped from the moving week.
3. The new, just completed, day is added to the moving week.
4. The new moving week is evaluated.

This approach has a number of advantages. First, all profiles will be evaluated within a maximum of one interval (currently one hour), so all recognizable events will be detected within that period. This evaluation interval can be shortened by resetting a 'granularity' parameter. Second, there is no discontinuity in the data being evaluated. Third, a history of past activity (at least a week) is maintained on-line. Fourth, the process lends itself well to near (within

the smallest interval) realtime processing. Fifth, the data-driven approach enables UNICORN to adjust easily to Cray down time or missing data.

2.2.4.2 Rule Development

An important first step in developing our expert rule set was interviewing the experts -- ICN security personnel. Interviews of administrators charged with establishing and enforcing the Laboratory's security policy were straightforward. The Laboratory has a well defined and documented security policy. Interviewing security auditors took time but was 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.

Another important part of our rule development was a statistical analysis of the audit record from the target Cray. We spent months reviewing the raw audit data. From this review we learned enough to implement an initial set of profiles, from which we calculated the characteristics of average UID and system behavior. We then studied those profiles that deviated significantly from the norm to determine which deviations comprised a suspicious event, particularly if combined with other indications.

This process of interviews and statistical analysis led to the definition of an initial rule set. We are now testing it against months of audit data. This testing phase will help us discover previously unidentified misuse scenarios and implement new rules to detect them. This process of testing and revising our rule set will be an ongoing one, as we continually aim to improve the accuracy of our system.

2.2.4.3 Rule Implementation

Expert rules are applied to the individual and composite profiles at the end of each interval, as described in Section 2.2.4.1. We have defined expert rules for three different intervals. *Hour rules* are applied at the end of each hour. *Day rules* are applied at the end of each hour, for the day thus far (one to twenty-four accumulated hours). *Week rules* are applied at the end of each day for the current running week (the current just-completed day plus the previous six days).

The server rule base comprises four logical rule filters; each designed to isolate certain types or levels of anomalous activities. We started by abstracting ICN security policy and well-defined invalid and suspicious behavior into rules that form the Primary Filter. Further refinements resulted in the Event Filter. Report requirements supplied rules for the Report Filter. The Alarm Filter determines the alerts resulting from each event. The server activates the rule base filters in order, as illustrated in Figure 2-2.

- The **Primary Filter** applies rules to the profiled data. These rules are straightforward descriptions of simple activities, each serving to distinguish a separate feature of anomalous behavior. The Primary Filter applies these rules individually; it does not correlate one with another. It assigns a Level-of-Interest to each anomaly defined by these rules. The results of this analysis are stored in the Report Table.
- The **Report Filter** applies rules to the anomalies output by the Primary Filter, to produce routine reports of anomalous behavior.
- The **Event Filter** applies rules to the anomalies identified by the Primary Filter. These rules try to identify patterns of anomalous activity that have a good chance of being systematic misuse (events). They specify what action to take when events are found, such as the scheduling and content of warning messages. The results of this analysis are stored in

the Event Table. Each event remains 'active' in the Event Table until security auditors resolve it off-line. Then it is flagged 'inactive' by the auditors. Inactive events are flushed from the table at regular intervals.

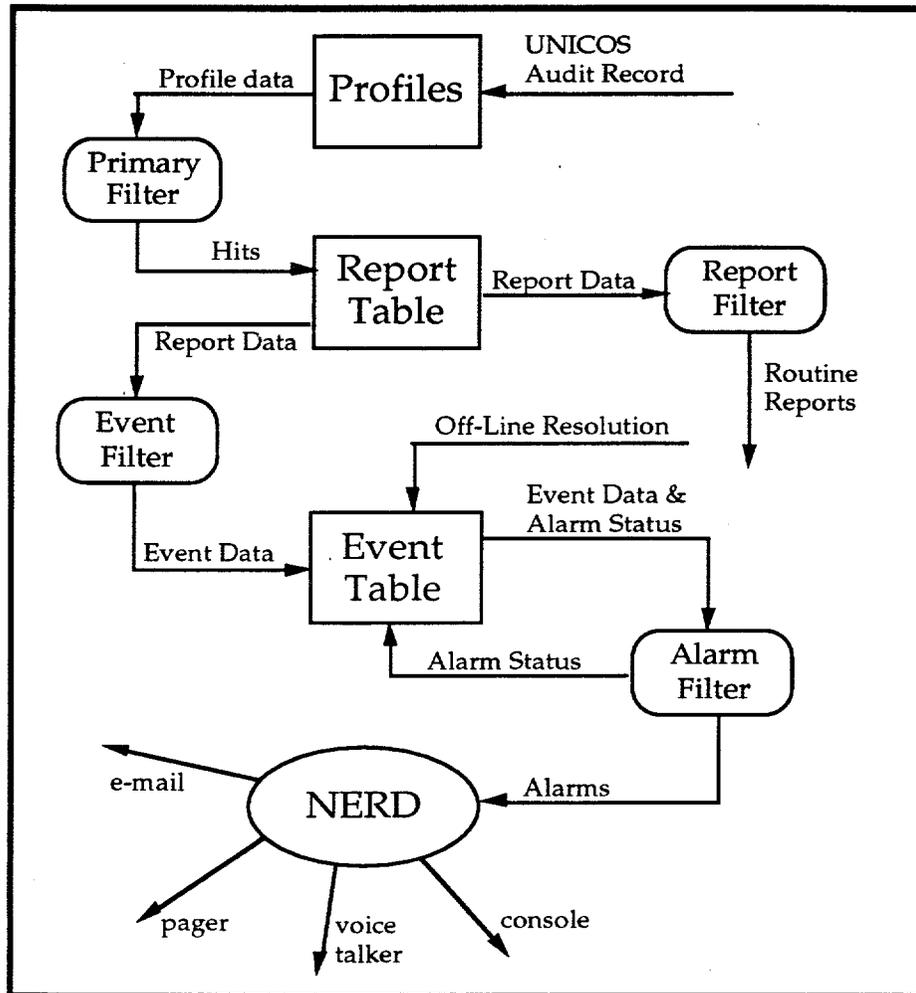


Figure 2-2: Expert Rule Implementation

- The Alarm Filter applies rules that manage appropriate notification of urgent or critical anomalous activity. It determines what level of alarms should be sent, and to who, and manages their frequency.

We encode our expert rules in a condition-action (if-then) form. The condition (if) describes a suspicious profile scenario or a violation of security policy. The action (then) specifies setting a level of interest for the relevant user (or composite user) profile. Table 2-10 gives an example of one complete rule. This rule focuses on the ratio of logon failures to total logons. Variable definitions are not included because UNICORN rule specifics are sensitive. Currently, the rule base consists of thirty-six such rules.

2.2.5 Reports

The server can report detected activity in several ways, including scheduled routine reports and, where required, immediate alarms. It also supports ad-hoc investigations, during which it can provide detailed reports of raw or profiled data in response to auditors' specific queries.

Table 2-10: RULE-CU-A-006 (Failure ratio)

<p>IF end of an hour $ctot_hour_logons = tot_csucc_logons + tot_cfail_logons$ $current_ratio = tot_cfail_logons / (ctot_hour_logons)$ THEN IF ($current_ratio$ is $> case_{n_min}$ AND $\leq case_{n_max}$) THEN IF $ctot_hour_logons > ctot4_{n_hour_max}$ THEN Set Rule CUH-006 to 4 in the Report_Table ELSE IF $ctot_hour_logons > ctot3_{n_hour_max}$ THEN Set Rule CUH-006 to 3 in the Report_Table ELSE IF $ctot_hour_logons > ctot2_{n_hour_max}$ THEN Set Rule CUH-006 to 2 in the Report_Table ELSE IF $ctot_hour_logons > ctot1_{n_hour_max}$ THEN Set Rule CUH-006 to 1 in the Report_Table EXPLANATION: Greater composite failure ratio than is normal for the previous hour.</p>
<p>IF end of an hour $ctot_day_logons = tot_csucc_logons + tot_cfail_logons$ $current_ratio = tot_cfail_logons / (ctot_day_logons)$ THEN IF ($current_ratio$ is $> case_{n_min}$ AND $\leq case_{n_max}$) THEN IF $ctot_day_logons > ctot4_{n_day_max}$ THEN Set Rule CUD-006 to 4 in the Report_Table ELSE IF $ctot_day_logons > ctot3_{n_day_max}$ THEN Set Rule CUD-006 to 3 in the Report_Table ELSE IF $ctot_day_logons > ctot2_{n_day_max}$ THEN Set Rule CUD-006 to 2 in the Report_Table ELSE IF $ctot_day_logons > ctot1_{n_day_max}$ THEN Set Rule CUD-006 to 1 in the Report_Table EXPLANATION: Greater composite failure ratio than is normal for the day thus far.</p>
<p>IF end of a day $ctot_week_logons = tot_csucc_logons + tot_cfail_logons$ $current_ratio = tot_cfail_logons / (ctot_week_logons)$ THEN IF ($current_ratio$ is $> case_{n_min}$ AND $\leq case_{n_max}$) THEN IF $ctot_week_logons > ctot4_{n_week_max}$ THEN Set Rule CUW-006 to 4 in the Report_Table ELSE IF $ctot_week_logons > ctot3_{n_week_max}$ THEN Set Rule CUW-006 to 3 in the Report_Table ELSE IF $ctot_week_logons > ctot2_{n_week_max}$ THEN Set Rule CUW-006 to 2 in the Report_Table ELSE IF $ctot_week_logons > ctot1_{n_week_max}$ THEN Set Rule CUW-006 to 1 in the Report_Table EXPLANATION: Greater composite failure ratio than is normal for the current week (the last seven days).</p>

2.2.5.1 Immediate Reports

Critical events are reported when they are detected. These are events that require prompt investigation. The server assigns a priority to each event, depending on its criticality. It then outputs an announcement to the UNICORN console and notifies a dedicated ICN system whose function is to log and report events for the entire ICN. This system is the Network Events Recording Device, or NERD [11]. The NERD provides four levels of notification; a broadcast using synthesized speech, paging, e-mail, and console display. The NERD undertakes appropriate notification based on priority, responsible individuals, and other information supplied by UNICORN.

2.2.5.2 Scheduled Reports

The server routinely generates reports every day. These reports cover the just-completed day and the just-completed running week (the just-completed day and the prior six days). These reports are transmitted to authorized personnel and stored electronically. Hardcopy summary reports are output once a calendar week.

The daily reports consist of a one-page activity summary, e.g., the number of active UIDs during the report interval, and the number of successful and unsuccessful user requests during that interval. There is also a set of graphs of different types of activity, plotted over time with a granularity of one hour. These are useful for representing abnormal patterns, such as an unusual spurt of off-hour usage. The rest of the report summarizes the results of the expert rule analysis. It lists suspicious UIDs in descending priority order (from the most suspicious to the least), with a list of the rules each has triggered. Finally, it lists all current (unresolved) events of interest, along with a list of events resolved during the report interval.

To support investigator follow-up, the server also produces a more detailed daily report that includes all raw data from the audit record. This data is the unprocessed audit record as received from the monitored Cray. Auditors occasionally need to review this data while attempting to ascertain what has happened during an event.

The server stores these regularly scheduled reports in a secure portion of our permanent file storage, where they can be accessed and reviewed only by authorized personnel.

2.2.5.3 Ad-Hoc Reports

The server can produce reports on demand. On-the-spot reports have proved invaluable in analyzing ongoing events. Finally, we use raw or profiled data that the server has saved to permanent file storage to perform ad-hoc background analyses of current and past activity. Authorized security personnel can examine this data using Sybase's built-in facilities, or pipe data to a statistical software package for more detailed analysis.

2.3 Off-Line Activities

Every day, security auditors review that day's report, and the current running week's report. When required, they review immediate alarms. They examine each anomalous event and decide whether to investigate it further. They analyze user or system audit data and may interview indicated users. An investigation may result in a warning to a user, or the user losing, at least temporarily, their ICN privileges. More often, it results in a learning experience for the user. The auditors file a short report at the completion of each investigation, giving details of its resolution. These reports, and periodic reviews of UNICORN by the security auditors, provide valuable feedback from which we continually try to improve the system. User response to these investigations has been surprisingly positive.

2.4 Data Integrity

We take care to protect the integrity of the Cray audit record throughout its life span. We treat it as sensitive because of its importance to security and accounting, and because its integrity is critical to ensure the validity of the intrusion and misuse detection process. Only a small set of system managers have access to the audit record on the Crays, in the file storage archive, and throughout the process of transmitting and analyzing it. We keep audit records in a secure part of the ICN, transmit them over secure lines, and backed them up routinely. Only authorized security auditors may examine any portion of the data or the reports generated by

UNICORN. We treat the results of investigations as sensitive. Such management activities are essential to the integrity of, and user trust in, the whole audit process [9].

3. Future Directions

UNICORN is nearing the end of Phase 1 development, at which point it will perform all the basic functions described in this paper. However, we expect to spend another year expanding this start into an optimally effective misuse detection system. During this year we expect to:

- expand collection and analysis to all pertinent data in the Security Log
- expand collection and analysis to additional logs (e.g., process accounting logs, sulog, tcp/ip logs)
- expand on-line vulnerability checks and misuse probes
- complete a user-friendly graphical user interface for investigative personnel
- provide near realtime notification of critical events using the NERD
- explore and perhaps implement active responses to critical events

Another future goal is to explore the possibility of supplementing our expert rulebase with a true anomaly detection component that "learns" typical behavior for each user, then reports deviations from these norms. Anomaly detection may also be applied to system-wide activity.

Acknowledgments

We acknowledge with gratitude the contributions of Jimmy McClary, who introduced us to the basic concepts of misuse detection, obtained our initial funding for NADIR, and supported us throughout the various incarnations of the project. We are indebted to Sharon Wilhelmy, who has reviewed NADIR's reports for three years, and who thus has been a valuable source of feedback on NADIR's functioning. As a result of her experience with NADIR, Sharon has been instrumental in helping us design UNICORN in a way that maximizes its usefulness to Los Alamos security auditors. We thank Steve Smaha (of Haystack Laboratories, Inc., Austin, TX) for his suggested standard audit record format. The format of the canonical UNICORN audit record was derived directly from this standard. We thank Judy Hochberg for her valuable contributions to the final editing of this paper.

References

- [1] R. Baldwin. *Rule-Based Analysis of Computer Security* (Massachusetts Institute of Technology, June 1987)
- [2] K. Jackson. *Development and Analysis of User Authentication Profiles for an ICN Intrusion Detection System* (Los Alamos National Laboratory, Technical Report, June 1989)
- [3] *SPI - Security Profile Inspector, Installation and User's Manual* (Lawrence Livermore National Laboratory, 1989)
- [4] D. Farmer, E. Spafford. *The COPS Security Checker System* (Proceedings of the Summer Usenix Conference, June 1990)

- [5] 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)
- [6] 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)
- [7] J. Hochberg, K. Jackson, J. McClary, D. Simmonds, *Addressing the Insider Threat* (Proceedings of the United States Department of Energy Computer Security Group Conference, May 1993, LA-UR-93-1181)
- [8] J. Hochberg, K. Jackson, C. Stallings, J. McClary, D. DuBois, J. Ford. *NADIR: An Automated System for Detecting Network Intrusion and Misuse* (Computers and Security, Elsevier Science Publishers Ltd., Volume 12, Number 3, May 1993, LA-UR-93-137)
- [9] K. Jackson, *Management Issues in Automated Audit Analysis: A Case Study* (Proceedings of the 8th European Conference on Information Systems Security, Control, and Audit, September 1993, LA-UR-93-2520)
- [10] S. Smaha. *A Common Audit Trail Interchange Format For UNIX* (Haystack Laboratories, Inc., Technical Report, May 1994)
- [11] D. Simmons. *Network Event Recording Device: An Automated System for Network Anomaly Detection and Notification* (Los Alamos National Laboratory, Technical Report, August, 1994)