Machine-Independent Audit Trail Analysis – A Decision Support Tool for Continuous Audit Assurance

---

Peter J Best

Associate Professor
School of Accountancy
Queensland University of Technology
Brisbane Australia
Email: p.best@qut.edu.au

George Mohay

Professor and Head
School of Computer Science
Queensland University of Technology
Brisbane Australia
Email: g.mohay@qut.edu.au

Alison Anderson

Senior Lecturer
School of Information Systems
Queensland University of Technology
Brisbane Australia
Email: a.anderson@qut.edu.au

Abstract

This paper reports the results of a research project which examines the feasibility of developing a machine-independent audit trail analyser (MIATA). MIATA is a knowledge based system which performs intelligent analysis of operating system audit trails. Such a system is proposed as a decision support tool for auditors when assessing the risk of unauthorised user activity in multi-user computer systems. It is also relevant to the provision of a continuous assurance service to clients by internal and external auditors.

Monitoring user activity in system audit trails manually is impractical because of the vast quantity of events recorded in those audit trails. However, if done manually, an expert security auditor would be needed to look for 2 main types of events - user activity rejected by the system's security settings (failed actions) and user's behaving abnormally (e.g. unexpected changes in activity such as the purchasing clerk attempting to modify payroll data). A knowledge based system is suited to applications that require expertise to perform well-defined, yet complex, monitoring activities (e.g. controlling nuclear reactors and detecting intrusions in computer systems).

To permit machine-independent intelligent audit trail analysis, an anomaly-detection approach is adopted. Time series forecasting methods are used to develop and maintain the user profile database (knowledge base) that allows identification of users with rejected behaviour as well as abnormal behaviour. The knowledge based system maintains this knowledge base and permits reporting on the potential intruder threats (summarized in Table 1).

The intelligence of the MIATA system is its ability to handle audit trails from any system, its knowledge base capturing rejected user activity and detecting anomalous activity, and its reporting capabilities focusing on known methods of intrusion. MIATA also updates user profiles and forecasts of behaviour on a daily basis. As such, it also 'learns' from changes in user behaviour.

The feasibility of generating machine-independent audit trail records, and the applicability of the anomaly-detection approach and time series forecasting methods are demonstrated using three case studies. These results support the proposal that developing a machine-independent audit trail analyser is feasible. Such a system will be an invaluable aid to an auditor in detecting potential computer intrusions and monitoring user activity.

KEY WORDS: Audit trails, intrusion detection, continuous assurance.

1.0 INTRODUCTION

Opportunities for new assurance services in an information technology environment have been recognised by the AICPA Special Committee on Assurance Services (SCAS) [AICPA (1996)]:

> Assurance services are independent professional services that improve the quality of information, or its context, for decision makers. It follows from this definition that new service opportunities must arise from the changing needs of decision makers. Information technology is changing those needs and therefore creating opportunities for new services.

The report produced by SCAS also forecasts a change in the timeliness of assurance services. It recognised that users may require audits of data on a "continuous basis" or on a "just-in-time basis", depending on the needs for decision making. Consequently, an evolution in information system assurance services is predicted to match these needs. Research interest in continuous auditing and assurance has been growing steadily over the past few years [Alles, Kogan & Vasarhelyi (2002); Rezaee et.al. (2002).

This paper deals with the feasibility of developing a knowledge based system to support a continuous assurance service. This research assesses the feasibility of developing a machine-independent audit trail analyser (MIATA) for on-line, multi-user, computer systems, which will support continuous auditing of computer user activity.

Threats to on-line systems, referred to herein as 'intrusions', have attracted considerable attention in the research literature [Denning et. al. (1987); Smaha (1988); Liepins & Vaccaro (1989); Lunt et. al. (1990); Winkler & Landry (1992); Lunt (1993), Mounji & Le Charlier (1997); Asaka (1999); Daniels & Spafford (1999)], Ye & Chen (2001), and Ye, Ehiabor & Zhang (2002). Prior research has focused on exploration of various approaches (eg. Pattern matching, stochastic models) to intrusion detection and their application to specific computer platforms (eg. Unix, Unisys). This paper contributes to this literature by:

1. Proposing a machine-independent approach to intrusion detection using a knowledge based system (KBS) for analyzing audit trails;
2. Demonstrating the feasibility of generating machine-independent audit trail records from disparate computer environments;
3. Demonstrating the suitability of an anomaly-detection approach to intrusion detection using time series methods of forecasting; and
4. Illustrating that data structures for MIATA can be developed imposing minimal storage requirements.

2.0 BACKGROUND

This research is concerned with threats to the availability, confidentiality and integrity of data [OECD (1992); NCSC (1985)]. On-line computer systems permit remote users to access programs and data. In such an environment, there is always the threat that users may obtain access to the private resources of other users and that intruders may infiltrate the system and obtain unauthorised access to both public and private resources. Loss of availability, confidentiality or integrity may result from such intrusions.

Various methods of intrusion have been used to obtain unauthorised access to on-line systems [Stoll (1988); Spafford (1989); Lunt (1993); Pfleeger (1997) and Proctor & Byrnes (2002)]. These methods include: 'attempted break-ins', 'masquerading', and 'browsing'. **Attempted break-ins, or password guessing** involves the intruder attempting repeatedly to guess a target user's password. With **masquerading**, the intruder logs in to the system as a target user, using his or her user identification (user-id) and password. The intruder may act in the name of the target user, accessing available files and programs. Where the target user has supervisor privileges, the intruder may use system utilities to modify the security characteristics of the system, such as adding new users, changing passwords and inflating privileges. **Browsing** refers to attempts by authorised users to obtain private information (such as user-ids) to assist in the above methods, or to perform unauthorised functions, such as accessing sensitive data files, changing user privileges, printing/displaying large numbers of files and resource hogging.

Four main safeguards against computer intrusions can be distinguished [Pfleeger (1997); Allen (2001); Krause & Tipton (2001) and Proctor & Byrnes (2002). **Authentication** is a mechanism for determining whether a user is authorised to use the system. Authorisation is given to users to enter the system in the form of a user-id and password. The user-id identifies the user. The password is a means of authenticating the user's identification. Other authentication techniques are available including biometric methods which recognize personal characteristics such as hand geometry and eye retinas. **Access Control** mechanisms aim to ensure that a user performs only authorised activities once granted access to the system. Each requested action by a user involving an object (file, program) is compared with an authority database. This database can be visualised as an "access control matrix" which specifies each user's privileges (owner, read, write, execute) in relation to particular objects. **Cryptography** involves encoding data into a form preventing outsiders from understanding their content if unauthorised access is gained. Cryptography can be applied to the protection of data in files, especially password files, and messages being transmitted across networks. **Audit Trail Analysis** can provide a powerful tool for detecting unauthorised activity and anomalous user activity. The above intrusion methods may leave evidence recorded in these audit trails of events such as repeated failed logins, logins at unusual times, changes in patterns of workstation, file and command usage by individual users and across the whole system, and repeated unsuccessful attempts to access files and execute commands, particularly those of a sensitive nature. A knowledge based system for audit trail analysis can provide the basis for a continuous audit assurance service by internal and external auditors by directing attention to potential intrusions and unusual changes in user activity.

3.0 A KNOWLEDGE BASED SYSTEM FOR AUDIT TRAIL ANALYSIS

The study of knowledge based systems has emerged from the field of artificial intelligence and is concerned with the investigation of methods and techniques for constructing computer systems that possess problem-solving expertise in specialised domains. Such systems have been developed over the past twenty years for a range of scientific (eg. medical and engineering) and business (eg. taxation and audit) applications [see Connell (1987); Abdolmohammadi (1987); Denning (1987); Quinlan (1989); Hellerstein, Klein & Milliken (1990); van Dijk & Williams (1990); Garcia & Chien (1992); Jagannathan et. al. (1993)]; and Tzafestas (1997)]. Typically, a KBS application is designed to make expertise (from one or more experts) in a given domain available to users (both experts and non-experts) at distributed locations in a consistent, uniform manner. Such applications may also be designed to handle well-defined, yet complex, monitoring activities (eg. controlling nuclear reactors). Intrusion detection in computer systems is a domain well-suited to a KBS solution, as discussed below.

Audit trail records typically provide a very low-level description of system activity. Reporting facilities for audit trails are usually very limited and 'instance-based'. These facilities allow an auditor to generate reports of all instances of specified activities. For example, all operator actions, all accesses to the payroll master files, or all failed logins can be reported. However, the opportunities for detecting intrusions are very limited.

How would one detect password guessing? A particular user's account may have been targeted by an intruder. Knowing the target's user-id, the intruder may attempt to log-in using the target's identification and then guess the password. Such a break-in attempt could be detected by identifying all users that had any failed logins in a given day, or (since users frequently miskey their passwords) an anomalous number of failed logins. In addition, such attempts may be perpetrated using a workstation different from that accessed normally by the target. Selecting users with anomalous failed logins and anomalous workstation usage would identify quickly users that may have been targeted for password guessing. It is clear that these simple notions are beyond the scope of instance-based reporting facilities which can only provide lists of events that meet specified criteria. These products provide no capability for relating a user's activity on a given day to the user's historical behaviour. A knowledge based system, however, can maintain individual user profiles or forecasts of user behaviour that allow the detection of such anomalous behaviour.

INSERT FIGURE 1 ABOUT HERE

A model of the processing system underlying MIATA is illustrated in Figure 1. Audit trail records are generated by the target system providing a record of successful and failed user activity - login, file access, command execution, etc. A machine-dependent preprocessor reads the audit trail records and generates records in a normalised, machine-independent format, herein referred to as 'normalised audit event' (NAE) records. A profile database is maintained that represents forecasts of expected user behaviour, based on actual behaviour and prior forecast errors. The machine-independent audit trail analyser (MIATA) reads NAE records, records daily activity, compares actual activity with forecasts, provides a user interface that identifies anomalous activity and potential intrusions, and updates profiles.

A KBS designed to analyse audit trails offers significant advantages. Such a system can facilitate the effective investigation of user activity. The system can also increase the sophistication of the analysis that can be performed routinely. The constraints in audit trail analysis of handling large volumes of low-level descriptions of activity using instance-based reporting facilities can be overcome using such a system. Combinations of activity, frequency of activity and unexpected user behaviour can be analysed, and attempted and successful intrusions can be detected. This level of software support for the auditor permits the development of a framework for audit trail analysis that can be applied consistently on a daily basis. Such analysis could identify the need for user training and could encourage an installation to develop explicit security policies concerning acceptable and unacceptable user behaviour.

A machine-independent KBS for audit trail analysis can be extremely valuable to organisations with multi-platform environments. Such organisations need an organisation-wide audit trail analysis solution. A machine-independent tool can provide a single, consistent method for analysing audit trails

from different systems and facilitate centralised control over computing resources [Rezaee et. al. (2002)].

4.0 PRIOR RESEARCH

The foundation paper in the area of audit trail analysis and intrusion detection, was Denning (1987). This paper introduced a number of concepts which are fundamental to the development of MIATA. Denning (1987: 223) developed an intrusion-detection model, consisting of six main components: **subjects -** initiators of activity on a target system - normally users; **objects -** resources managed by the system - files, commands, devices, etc.; a**udit records** – generated by the target system in response to actions performed or attempted by subjects on objects - user login, command execution, file access, etc.; **profiles -** structures characterising the behaviour of subjects with respect to objects in terms of statistical metrics and models of observed activity; **anomaly records**: -generated when abnormal behaviour is detected; and **activity rules** - actions taken when some condition is satisfied, which update profiles, detect abnormal behaviour, relate anomalies to suspected intrusions, and produce reports.

Denning (1987: 224-226) proposed alternative methods for developing activity profiles. Such profiles characterise the behaviour of a given subject with respect to a given object, representing a definition of normal activity and a means of detecting anomalous activity. Observed behaviour can be characterised in terms of statistical metrics (such as event counters, interval timers, and resource measures) and models (such as operational models that compare observations against fixed thresholds, mean and standard deviation models that define an observation as abnormal if it falls outside a confidence interval that is a specified number of standard deviations from the mean, multivariate models that incorporate correlations among two or more metrics, markov process models, and time series models).

Early audit trail analysers have been designed to analyse audit records of a single computer system environment. Examples include the Standard Audit File of & Pinnis (1984), SRI's early model of the Intrusion Detection Expert System (IDES) [Denning et. al. (1987); and Lunt et. al. (1988)], Haystack Laboratories' HAYSTACK System [Smaha (1988)], Los Alamos National Laboratory's Wisdom & Sense (W&S) [Liepins & Vaccaro (1989)], AT&T's COMPUTERWATCH [Dowell & Ramstedt (1990)] and ASAX [Habra et. al. (1992)]. These projects have focused on exploration of various approaches (eg. pattern matching, stochastic models) to intrusion detection and their application to specific computer platforms (eg. Unix, Unisys and VMS). Many of these approaches have been applied to networks of computers [Lunt et. al. (1990); Snapp et. al. (1991); Winkler & Landry (1992); Jagannathan et. al. (1993); Mounji & Le Charlier (1997); Daniels & Spafford (1999); and MIT's DARPA Intrusion Detection Evaluation Project (http://www.ll.mit.edu/IST/ideval/index.html) [Haines, et. al. (2001)]. These papers provide little justification for nor detailed information on the statistical methods used. Little documentation is provided of attempts to analyze actual user audit trails in order to justify and test the methods selected.

This research can be distinguished from earlier research in several ways. A machine-independent approach to intrusion detection is proposed using a knowledge based system (KBS) for analyzing audit trails. The feasibility of generating machine-independent audit trail records from disparate computer environments is demonstrated. Case studies of actual user activity are used to demonstrate the suitability of an anomaly-detection approach to intrusion detection. Time series methods of forecasting produce results satisfying 'goodness of fit' requirements. This research also illustrates that data structures for MIATA can be developed imposing minimal storage requirements.

5.0 MACHINE-INDEPENDENT AUDIT TRAIL RECORDS

The existence of an audit trail mechanism in computer systems is necessary before intelligent audit trail analysis can be contemplated. The primary pronouncement guiding operating system development over the past fifteen years has been the *Trusted Computer System Evaluation Criteria* (TCSEC), [NCSC (1985)]. The NCSC, previously the Department of Defense (DoD) Computer Security Center, developed the TCSEC to serve as a basis for rating computer systems. The TCSEC was published originally in 1983 and was adopted as a DoD Standard in 1985.

The TCSEC defines four divisions: D, C, B, and A, arranged in a hierarchy with division D referring to Minimal Protection and division A referring to Verified Protection. There are seven levels within these four divisions: D, C1, C2, B1, B2, B3 and A1. Each level includes all of the security provisions of the preceding levels. Accordingly, C2 security is greater than C1, and A1 is greater than all other levels. Pfleeger (1989: 283-287) summarises the distinctions between these ratings. Operating systems vary greatly in their security ratings. Early Unix systems would be rated at C1 [see Farrow (1991: 16], although more recent versions of Unix have achieved higher ratings [Santa Cruz Operation (1992: 268-287)]. The Honeywell Multics and Scomp operating systems have been rated B2 and A1 respectively [Pfleeger (1989: 286)]. TCSEC classes C2 through A1 have an audit trail requirement that a user's actions should be open to scrutiny by means of an audit.

Five purposes of the audit trail mechanism can be identified [Datapro (1988: 103)]. Auditors may review patterns of access by users to objects (eg. dial-up lines), discover attempts to bypass security, or to leverage privileges. Audit trails also can act as a deterrent against attempts to bypass security and provide evidence of system misuse.

The TCSEC class C2 is the first level with an audit trail requirement. The C2 requirements are common to all the levels above C2. In order to permit audits to be performed on the basis of individual users and objects, the following events are subject to audit at the C2 class [see Datapro (1988: 105)]: use of identification and authentication mechanisms, introduction of objects into a user's address space, deletion of objects from a user's address space, actions taken by computer operators and system administrators and/or system security administrators, and all security-relevant events. Security-relevant events are defined by the TCSEC as "*any event that attempts to change the security state of the system (e.g., change discretionary access controls, change the security level of the subject, change user password, etc.). Also, any event that attempts to violate the security policy of the system (e.g., too many attempts to log in, attempts to violate the mandatory access control limits of a device, attempts to downgrade a file, etc.)*".

The following data are to be recorded by the audit trail mechanism at the C2 class [see Datapro (1988: 105)]: date and time of the event, identification of the subject (eg. user-id), type of event, success or failure of the event, origin of the request (eg. workstation ID), name of object introduced, accessed, or deleted from a user's address space, and modifications made by the system administrator to the user/system security databaseds. Higher TCSEC classes require additional events to be audited [Datapro (1988: 105-6)].

This paper aims to demonstrate the feasibility of machine-independent audit trail analysis. Since operating systems with a TCSEC rating below C2 have no audit trail requirements, the minimal audit trail requirements of the C2 class is adopted as the basis for defining a Normalised Audit Event (NAE) (and corresponding record format) and for predicting profiles associated with particular intrusions.

Accordingly, this generalised representation of user activity is applicable to all operating systems with a TCSEC rating of C2 or higher, which include the majority of commercially-available operating systems. For the purposes of this study, each NAE record has the following attributes: date of event, time of event, identity of subject (eg. User-id), workstation identification, type of event, identity of the object (eg. File) and status of the event (success or failure).

In the MIATA system, a machine-dependent preprocessor is used to generate NAE records from the target system's audit trail records. MIATA reads NAE records and performs audit trail analysis, regardless of the target system's hardware/software environment. It is proposed also that intruder profiles for known methods of intrusion can be developed from such NAE records of user activity.

The characteristics of the known active methods of intrusion in terms of patterns of observed user behaviour may be identified in order to develop a set of intruder profiles. Each audit record identifies an event involving a specific subject and object combination. In addition, the status of each event is indicated in terms of success or failure. The success or failure of an event may be relevant to particular methods of intrusion. A specific method of intrusion may involve also an "anomalous" level of activity, that is, a frequency of a specific type of activity (eg. login) per time period that is abnormal for the subject in question, or differs significantly from forecasts of that activity for the subject. Characteristics of the three active methods of intrusion are discussed below:

1. **Password guessing**. The intruder tries repeatedly to guess a user's password. This method requires the intruder to be aware of an authorised user's identification (user-id). A wide range of direct methods of password attack have been identified [see Witten (1987)]. However, the effectiveness of such methods can be reduced severely where passwords are properly managed. Passwords may be obtained by using special software that tries thousands of combinations of characters. Where the target system has limits on the number of unsuccessful login attempts, this software can terminate the dial-up session, reconnect and resume the experiment automatically. Password guessing is characterised most likely by an anomalous number of failed login attempts by the user, usually associated with changes in workstation usage. In addition, anomalous levels of failed logins for a particular workstation or at the system level may indicate that password guessing is occurring.

2. **Masquerading as an authorised user**. The intruder may log in to the system employing the user-id and password of a target user, or issue commands by using a workstation tapped into the user's communication line. The masquerader may wish to access commands and files used frequently by the target user, or use this account as a spring-board into other parts of the system. Masquerading may be indicated by an anomalous number of logins for a user, since inactive users are often targeted and since the intruder is also logging-in to the system using the same user-id. The target user's workstation usage may appear anomalous, since it is unlikely that the intruder can obtain access to the user's normal workstation. The effectiveness of this means of detection depends on whether physical workstation addresses are employed on the system. The intruder may attempt to access the system at unusual hours. Masquerading may also be indicated by anomalous file and command usage by the target user (that is, changes in the user's patterns of file and command access), or anomalous levels of failed activity, since the intruder may attempt to exploit the user's privileges to its limits.

3. **Browsing**. A legitimate user attempts to access files or execute commands in an unauthorised manner. The user experiments with the system, testing his or her privileges to access files and commands of interest. Users may attempt also to misuse sensitive files and commands.

Browsing by a particular user is characterised most likely by anomalous file and command usage, anomalous failed file and command usage, and anomalous levels of failed actions.

Table 1 summarizes the likely activity profiles associated with the three main intrusion methods.

INSERT TABLE 1 ABOUT HERE

The intruder profiles developed above require MIATA to detect anomalous user activity. Accordingly, there are implications for the design of MIATA concerning what events need to be recorded in NAE records, what methods are suitable for detecting anomalous user activity and what data series need to be maintained and forecast.

The intruder profiles for password guessing, masquerading, and browsing require MIATA to detect anomalous activity by users with respect to logins, late logins, failed logins, failed actions, workstation usage, file usage, failed file access, command usage and failed command execution. The NAE record format incorporates: date of the event, time of the event, user-id, workstation identification, type of event, object-id, and status (success or failure). The types of events that need to be recorded in the NAE to permit audit trail analysis are: login; file creation; file deletion; file rename; file read; file write; and command execution.

These events must be extracted from the target system's audit trail records by the machine-dependent preprocessor. Login events will permit login frequency for each user to be computed by MIATA. The time stamp in the NAE record will permit MIATA to detect late logins. The status (success or failure) of an event will permit MIATA to distinguish successful and failed user logins and actions. The workstation identification will permit MIATA to monitor workstation usage by users. File- and command-related events will specify the particular object-id, thereby permitting MIATA to monitor file and command usage by users. Accordingly, sufficient data concerning these event types can be collected using NAE records to permit MIATA to detect intrusions and anomalous user activity.

An anomalous level of activity (eg. failed logins) is defined as a frequency of activity per day that is abnormal for the particular user. MIATA adopts a forecasting approach to detecting such anomalies. In particular, time series methods are evaluated to select a means of generating forecasts and confidence intervals for user activity. A computed frequency of activity will be categorised as anomalous if it lies outside the relevant confidence intervals. In addition, to permit MIATA to detect anomalous usage of workstations, files and commands, MIATA's user profile database must maintain sufficient data to permit forecasts to be generated for individual user's actions involving particular workstations, files and commands.

This discussion has focused on known methods of intrusions to computer systems in order to develop intruder profiles. These profiles rely extensively on the ability of audit trail analysis to detect anomalous activity by users. Concentrating on anomalous activity offers several advantages. System misuse may be detected. Acceptable and unacceptable changes in user interests and behaviour may be identified. Finally, intrusions may be discovered which are novel, that is do not conveniently fall into one of the above-described categories.

6.0 ANOMALY DETECTION USING FORECASTING METHODS

Earlier audit trail analysers have adopted a wide range of statistical approaches for detecting anomalous activity by users. Javitz et. al. (1993: 1-25) provides an outline of several alternative statistical approaches that may be applicable for anomaly detection. Ongoing research in these areas has explored the application of stochastic models, chi-square statistics, cluster-based learning and fuzzy rules to intrusion detection [Luo & Bridges (2000); Ye & Chen (2001); Li & Ye (2002) and Ye, Ehiabor & Zhang (2002)]. The MIATA system uses forecasting methods as an efficient approach for anomaly detection.

6.1 FORECASTING METHOD SELECTION

The fundamental proposition for the success of the MIATA project is that it is feasible to forecast user behaviour and to employ such forecasts for categorising user behaviour as normal or anomalous.

Makridakis & Wheelwright (1979) provides a simple framework for classifying forecasting methods. Two dimensions are utilised for describing forecasting situations - the type of pattern experienced and the type of information available. The type of pattern experienced may be one where history is expected to repeat itself, that is where the historical pattern is expected to continue into the future, or one where the pattern depends on external factors as well as historical observations. In certain forecasting situations, quantitative historical data may be available, such as sales quantities. In other situations, only qualitative data may be available, such as executive opinion.

Audit trail analysis is a situation where quantitative information is available. The MIATA system incorporates components which record the frequency of user actions and maintain user profiles which consist of forecasts of the frequency of user actions on a daily basis. Accordingly, the quantitative forecasting methods appear most appropriate for implementation in MIATA.

It is expected that user behaviour will prove to be relatively stable. Users who have been assigned specific job descriptions or roles in commercial organisations are likely to have relatively stable patterns of system usage. They are likely to use primarily the same workstations, and access primarily the same files and programs each day in order to perform their assigned functions. A portion of a user's system usage may in fact involve scheduled batch jobs. It is acknowledged that there may be minor random deviations from this stable pattern and that there may be routine deviations on a periodic basis, such as at the end of the working week, at the end of the month, and at the end of the financial year.

In addition, the privileges assigned to specific users should also reflect their roles. Linked to the principle of *separation of duties* (NCSC 1985), the security principle of *least privilege* requires that a user should have access to the fewest objects (files and programs) needed to perform the functions associated with his/her role. An organisation's security system is likely to be most effective if these principles are implemented and if so, they also have the effect of stabilising the activity of individual system users. Given these arguments, it is proposed that basing forecasting method selection on the assumption that the pattern of historical behaviour will repeat itself is quite supportable.

Explanatory or *causal* methods such as regression assume that the factor to be forecast exhibits a cause-effect relationship with one or more independent variables. Such methods determine the form of this relationship and use it to forecast future values of the dependent variable given observations of the independent variable(s). In contrast, *time series* methods assume that future values of a factor can be predicted merely from historical values of the factor and/or past forecasting errors. Such methods determine the pattern in these past values and extrapolate that pattern into the future.

In the current context, time series methods offer a practical advantage over explanatory or causal methods. In order to generate a forecast, the system requires access only to historical values and/or errors. For the MIATA project, the selection of suitable time series forecasting methods should consider the nature of the forecasting situation and the characteristics of the alternative methods. The MIATA project involves the generation of immediate time horizon forecasts, that is, daily forecasts of the frequency of user activity. Such a time horizon reduces severely the impact of data patterns with trend, seasonal or cyclical characteristics. Because users are performing stable external organisational roles, it is expected also that case studies of actual user behaviour will demonstrate the relative stationarity of the data series. Superior accuracy may be achievable using more sophisticated methods such as decomposition and ARMA. However, the forecasting situation faced is one where tens of thousands of items must be forecast on a daily basis, where data storage requirements must be minimised, where development and running costs must be minimised and where complexity must be kept to a reasonable level.

Given the above arguments, only simple smoothing methods were recommended for potential incorporation in MIATA. These methods are incremental in nature, in that they generate forecasts from prior forecasts and errors. Such methods are suitable for immediate time horizon forecasting for relatively stationary data series, and impose minimal costs in terms of development, data storage and running costs.

6.2 THE FEASIBILITY OF FORECASTING USER ACTIVITY

The objective of this experiment was to compare the performance of simple smoothing time series methods in forecasting actual user behaviour. This section describes a case study aimed at comparing the effectiveness of the selected forecasting methods.

Forecasts were generated using each of six time series methods for each data series. The accuracy of each of five smoothing methods - adaptive-response-rate single exponential smoothing (ARRSES), Brown's one-parameter linear exponential smoothing (BROWN), simple average (MEAN), single moving average of order 5 (MOVAV), and single exponential smoothing (SES) - is evaluated relative to each other and to the naive method (NAIVE). For each data series, the method which minimises forecasting error was identified. The error measures used in comparing methods were the Mean Squared Error (MSE) and the Mean Absolute Percentage Error (MAPE). The "goodness of fit" of the more accurate method was then assessed by examining the autocorrelation coefficients (ACFs) of the errors.

The case study employed data from a statutory rail authority’s system. The primary processing applications are payroll/personnel, wagon tracing, freight accounting, seat/berth reservations, crew rostering, locomotive maintenance and supply. There are over 5500 interactive users of the system employing over 1700 workstations.

A set of 20 users was selected for monitoring with the assistance of Information Systems Audit staff. Users were selected with high volumes of activity and a wide range of action-on-object events. Seven weeks' data was collected for the 20 users. After removing weekends and public holidays, user activity data for 32 weekdays was available for conducting the forecasting experiment. It was decided early to restrict these experiments to week-day data. Relatively few users were active weekend users.

Preliminary experiments indicated that better results would be obtained by operating two versions of MIATA, one to monitor week-day user activity and another to monitor weekend user activity

A total of 220 data series were chosen for testing the effectiveness of the selected forecasting methods. These data series focused on the frequency of individual user's actions on objects. Successful logins, late logins and failed logins were collected for each of the 20 users. Each user's failed actions data series was also collected. Another 44 data series dealt with individual users' workstation usage - successful and failed logins. Sixty-six (66) workstations were used by these 20 users over the 32-day period, thereby giving 132 (66 successful logins and 66 failed logins series) potential data series for analysis. User-workstation relationships were selected with at least 15 successful logins over the 32 days (22 of the 66 workstation usage relationships). Both successful logins and failed logins data series were analysed for these relationships.

The 20 users attempted to access (read, write, failed read or failed write) 321 files over the 32 days, giving 1284 potential data series. File usage data series were selected with at least 20 actions over the 32 days resulting in 89 data series for analysis. Only 19 different commands executed by these 20 users over the 32-day period were monitored by security audit software and recorded in the operating system audit trail. Accordingly, relatively few command usage data series were available for analysis. Of the 38 potential data series (executes and failed executes), only those with at least 5 actions over the 32 days were selected for analysis (7 data series).

The forecasting experiments were conducted using a spreadsheet - FORECAST. The FORECAST spreadsheet performs a simulation of the six forecasting methods for any given data series. The following parameters were used: ARRSES - $\alpha_1=.2$ and $\beta=.2$; BROWN - $\alpha=.1$; SES - $\alpha=.2$. FORECAST simulates the generation of the daily forecasts for each of the six methods, measuring daily forecast errors under each method, and printing a summary report. This report lists the data series with the resulting forecast errors and summarises the MSE and MAPE for each method. This output allows the identification of the method which performs best for each data series, on the basis of each measure of forecasting accuracy, and provides the error data to permit an analysis of the "goodness of fit" for the superior method. FORECAST also provides the facility to display or print a graphical representation of the performance of the methods for a given data series.

Each data series was subdivided into an initialisation set of 5 observations and a test set used in comparing forecasting accuracy on the basis of MSE and MAPE. The mean method minimised MSE and MAPE for the greatest proportion of the data series. This result suggests that these data series exhibit a high degree of stationarity and provides support for the focus on smoothing methods.

The five smoothing methods outperformed the naive method in terms of minimising MSE and MAPE. The measures of MSE and MAPE for the smoothing methods did not vary greatly for individual data series, causing a significant number of ties when the performance of these methods was ranked. Seventy-nine percent (173) of the data series had a MSE range (maximum MSE less minimum MSE among the smoothing methods) for the 32 days of less than 5, causing little variation over the five methods in *standard error*, that is, less than 2.2, and little variation in associated confidence intervals for a data series among these methods. (MIATA uses each data series' standard error to derive confidence intervals).

The randomness of the mean method's forecasting errors was examined in order to test its "goodness of fit". For 95 percent confidence, all ACFs should lie within the range (where $n=31$):

(1) $I_{95\%} = 0 \pm \frac{1.96}{\sqrt{n}}$

$= 0 \pm .352$

Seventy-six percent (167) of the 220 ACF plots for the forecast errors produced by the mean method satisfied this criterion. The mean method provided a "good fit" for 76 percent of the data series. Fifty-three of the plots showed ACFs lying marginally outside the required interval. The histogram plots for the forecast errors were also examined for consistency with a normal distribution with a zero mean and absence of consistent observable skewness. Eighty-five percent (188) of the histograms appeared normal. Some minor skewness was apparent in 32 of the plots.

Similar results were obtained with two further case studies. These studies involved two different organisations (a government service bureau and a tertiary institution) with different computing environments. While not conclusive, these results provide support for the application of smoothing time series methods to forecasting computer user behaviour and suggest that such forecasts (with their associated confidence intervals) can be employed to successfully categorise user behaviour as normal or anomalous.

A significant number of data series were excluded from forecasting experiments in the case studies. These data series could be categorised as *sparse*, in that they involved few (such as less than 5) observations over the 32-day period. These data series had zero observations on at least 27 of the 32 week-days. Some comments are provided below concerning the significance of these exclusions and the performance of the naive and smoothing forecasting methods on these data series.

Sparse data series, such as failed logins or failed writes, involve a large number of zero observations and typically a few non-zero observations of low magnitude. Accordingly, forecasts and standard errors (and associated confidence intervals) tend to be close to zero. Minimal variation in forecasting accuracy is observable among the six methods. The smoothing methods however appear to consistently outperform the naive method. Since the majority of forecast errors (at least 27 of the 32) are close to zero, the requirement for the errors to be normally distributed around a mean of zero is satisfied by all methods. In cases where the nonzero observations are relatively close to zero, the ACFs satisfy the requirements for a good fit. The ACFs however may be distorted where the non-zero observations are large in magnitude.

It may be concluded therefore that the smoothing method selected for incorporation in the MIATA system on the basis of the above case studies is also suitable for forecasting and detecting anomalous levels of activity for these sparse data series. Non-zero observations on a particular day are likely to exceed their confidence intervals and be categorised as anomalies.

6.3 LIMITATIONS OF FORECASTING TECHNIQUES

A number of potential limitations can be identified where forecasting methods are used in intrusion detection systems. The effectiveness of the MIATA system in detecting intrusions depends on its ability to identify **anomalies**. MIATA's forecasting and anomaly detection component derives confidence intervals which are compared with actual observations. These confidence intervals are

determined with reference to the forecast (F_t), the standard error (SE_t) and a specified number of standard errors (δ).

The effectiveness of the MIATA system in detecting actual intrusion activity also depends on the characteristics of the user targeted by the intruder. Where the target's data series are relatively stable, the corresponding confidence intervals will be narrower than if those data series are more variable. Additional actions by the intruder who is masquerading as the authorised user may be reported or not reported as anomalous depending on the width of those confidence intervals. However, all activity attempted by the intruder which is new for the authorised user will be categorised and reported as anomalous.

MIATA's auditor reporting component focuses attention on users meeting the activity profiles associated with the intrusion methods and on users with larger volumes of anomalous actions. However, a more sophisticated intruder may be aware or cautious of potential monitoring activities and accordingly concentrate on low volume activity. Given the above discussion, such an intruder may target relatively busy users with more variable data series and avoid detection.

Genuine anomalies resulting from random events, special projects or end-of-period tasks may distort significantly the confidence intervals for particular data series. These confidence intervals may remain high for some time as a result and conceal potential intrusions. Ideally, the auditor could be capable of investigating each anomaly and providing input to the system to specify whether it should be incorporated in the data series as an observation or omitted to prevent distortion of confidence intervals. Such an approach is not considered feasible at this stage.

It is acknowledged that these results are based on samples of user activity over relatively short periods of time and are not necessarily generalizable to populations, such as all users or all computer installations. However, these encouraging results provide support for the proposition that computer user behaviour is forecastable. It is suggested that further case study research should be conducted to accumulate further compelling evidence supporting this proposition.

7.0 USER PROFILE DATABASES

This section examines MIATA's forecasting and anomaly detection component, and its data structures.

7.1 THE FORECASTING AND ANOMALY DETECTION COMPONENT

The MIATA system's forecasting and anomaly detection component performs three main functions on a daily basis - generating forecasts for each activity attribute of interest, detecting anomalous behaviour in relation to those forecasts, and updating profiles.

A forecast is generated for each *activity attribute* (such as successful logins) in the profiles. The empirical forecasting case studies documented above support the adoption of the mean method for incorporation in the MIATA system, imposing minimal storage requirements on the system. The *ForecastData* composite data element consists of five attributes – *NoToday*, *NoDays*, *Forecast*, *MeanSqError* and *AnomalyFlag*. The NAE recording component uses the *NoToday* attribute to accumulate daily activity frequency, for example, the total number of successful logins for a user today. The *NoToday* (X_t), *NoDays* (or number of observations - t) and *Forecast* (F_t) attributes provide sufficient data to generate the forecast (F_{t+1}) for the following day as follows:

$$(2) \qquad F_{t+1} = \frac{F_t\, t + X_t}{t+1}$$

The *MeanSqError* attribute measures the historical MSE for the particular activity and provides a basis for the generation of a *standard error* measure that is used in establishing confidence intervals around the new forecast. The *AnomalyFlag* attribute indicates the outcome of the anomaly detection process for a particular activity.

Anomaly detection proceeds once the NAE recording component has processed the current day's audit events. The anomaly detection process is distinct from the auditor reporting component of the MIATA system. Its purpose is to examine *ForecastData* for each activity attribute, establish appropriate confidence intervals around the forecast, and set the *AnomalyFlag* attribute to indicate anomalous activity where the *NoToday* value lies outside those intervals. The setting of the *AnomalyFlag* attribute allows the auditor to select various reporting alternatives. The anomaly detection process need be performed only once each day.

The anomaly detector applies the following decision rule to each activity attribute in the profiles:

If X_t lies within confidence intervals, conclude *No Anomaly*
If X_t lies outside confidence intervals, conclude *Anomaly*

Confidence intervals (C_t) are determined by reference to the *Forecast* (F_t) and the *MeanSqError* (MSE_t) attributes. The auditor sets the range of these intervals by specifying δ, the number of standard errors (SE_t), to be incorporated. [See Equation (3)]. To illustrate, if the distribution of observations is assumed to be normal, a value for δ of 3 defines confidence intervals that include 99.7 per cent of observations. Accordingly, an observation may be characterised as anomalous if it lies outside those intervals. By definition, new activity by a user is also characterised as anomalous since F_t and SE_t have zero values.

$$(3) \qquad C_t = F_t \pm \delta\, SE_t$$

where

$$SE_t = \sqrt{MSE_t}$$

Two types of errors are feasible with the above decision rule: (1) being led by the decision rule to conclude that an anomaly exists when in fact none does exist - a *Type I error*; and (2) being led by the decision rule to conclude that no anomaly exists when in fact one does exist - a *Type II error*. Generally, confidence intervals can be set to minimise either Type I or Type II errors, but not both. Increasing the value of δ widens confidence intervals and reduces the likelihood of incorrectly classifying observed behaviour as anomalous. However, it also increases the likelihood of failing to detect anomalous behaviour. The anomaly detection component permits the auditor to reduce the likelihood of Type I errors by increasing the value of δ. It could be argued that Type II errors, however, are the primary concern of the MIATA system - failing to detect anomalous activity. As a result, a more effective anomaly detection mechanism may be achieved by setting a more conservative δ value (such as 1) such that a greater proportion of observations are characterised as anomalous. In such a case, reporting facilities are required which allow effective use of the auditor's time in investigating flagged anomalies.

The anomaly detection process also maintains *ForecastData* for each user's *WorkstationAnomalies*, *FileAnomalies* and *CommandAnomalies*. This feature permits the focussing of attention on users with large numbers of anomalies, unusual numbers of anomalies for the particular users, as well as combinations of workstation-, file- and command-anomalies which may suggest intrusive behaviour.

The last function of this component of the MIATA system is the daily updating of profiles. A set of end-of-day procedures are required to reset anomaly flags, generate forecasts for the next day, and maintain the profiles. Records are removed that correspond to entities that have been deleted and inactive entities, that is, those that represent relationships between users and objects which have apparently been discontinued. The auditor may use utilities to monitor storage occupied by the profiles. This component of the MIATA system allows the auditor to specify the retention period for such inactive entities. This facility along with the selection of the forecasting method help minimise storage requirements.

7.2 DATA STRUCTURE

The MIATA system accumulates daily frequency data for users and objects, generates forecasts and detects anomalous behaviour. A logical data structure is required that focuses on individual users, individual objects and relationships between users and objects.

The logical data structure for the profiles in the MIATA system incorporates four principal entities - *User*, *Workstation*, *File* and *Command*. Each of these entities have an identifying attribute - a unique name - distinguishing the entity from others of the same type. The other attributes of these entities are primarily activity attributes represented by the ForecastData composite data element. Three additional entities are included in the data structure - *WorkstationUsage, FileAccess and CommandExec* - concerned with a user's workstation usage, file access and command execution respectively. A *SystemMetric* entity is also defined for maintaining activity data for the entire system, namely failed logins, failed reads, failed writes, failed executions and failed actions.

Figure 2 shows the data elements for the entities in the MIATA data structure. A distinct record type (and databased table) is required for each entity type. The key of the record type in each case is its identifying data element(s) (shown underlined). Data elements shown in bold are represented by the ForecastData composite data element and are composed of its simple component data elements.

INSERT FIGURE 2 ABOUT HERE

8.0 SUMMARY AND CONCLUSIONS

The results from this project support the proposal that developing a machine-independent audit trail analyzer is feasible. This MIATA project has demonstrated that machine-independent audit trails are feasible, that an anomaly-detection approach to intrusion detection using time series forecasting methods can produce acceptable results and that the data structures for MIATA system can impose minimal storage requirements. Such a system will be an invaluable aid to an auditor in detecting potential computer intrusions and monitoring user activity. The system will serve as an attention directing tool for the auditor faced otherwise with masses of low-level system activity data and limited reporting facilities.

Development of an effective tool for analysing audit trails can assist auditors in monitoring the security of their clients' systems. The audit trails provide a feedback mechanism on the operation of the security policies implemented on these systems. Without the availability of a monitoring device such as MIATA, the auditor's focus is merely on reviewing the preventive mechanisms for enforcing security. The MIATA system facilitates the development of a framework for audit trail analysis which can be applied consistently on a daily basis. Such an analysis may identify objects requiring better protection, users requiring disciplinary action and users needing training, and may suggest areas where improvements in the client's security policies are required. The MIATA system will be extremely valuable in organisations with multi-platform environments, since a single machine-independent tool can provide a consistent basis for analysing audit trails from different hardware environments.

BIBLIOGRAPHY

Abdolmohammadi, M.J. (1987) "Decision Support and Expert Systems in Auditing: A Review and Research Directions", *Accounting and Business Research*, Vol. 17, No. 66, pp. 173-185.

AICPA (1996) *Report of the Special Committee on Assurance Services,* AICPA.

Allen, J.H. (2001) *The CERT Guide to System and Network Security Practices*, Addison-Wesley, Boston.

Alles, M.G., Kogan, A. & Vasarhelyi, M.A. (2002) "Feasibility and Economics of Continuous Assurance", *Auditing: A Journal of Practice & Theory*, Vol. 21, No. 1, pp. 125-138.

Asaka, M. (1999) "Information Gathering with Mobile Agents for an Intrusion Detection System", *Systems and Computers in Japan*, Vol. 30, No. 2, pp 31-37.

Connell, N.A.D. (1987) "Expert Systems in Accountancy: A Review of Some Recent Applications", *Accounting and Business Research*, Vol. 17, No. 67, pp. 221-233.

Daniels, T.E. & Spafford, E.H. (1999) "Identification of Host Audit Data to Detect Attacks on Low-Level IP Vulnerabilities", *Journal of Computer Security*, Vol. 7, No. 1, pp. 3-35.

Datapro Research Corporation (1988) *Auditing Trusted Computer Systems: National Computer Security Center (NCSC),* Datapro Research Corporation, Delran, N.J.

Denning, D.E. (1987) "An Intrusion Detection Model", *IEEE Transactions on Software Engineering*, Vol. SE-13, No.2, pp. 222-232.

Denning, D.E., Edwards, D.E., Jagannathan, R., Lunt, T.F. & Neumann, P.D. (1987) *A Prototype IDES: A Real-Time Intrusion-Detection Expert System*, Technical Report, SRI International, Menlo Park, CA.

Dowell, C. & Ramstedt, P. (1990) "The COMPUTERWATCH Data Reduction Tool", *Proc. 13th National Computer Security Conference*, Baltimore, MD, October, pp. 99-108.

Garcia, O.N. & Chien, Y.T. (1992) *Knowledge-Based Systems: Fundamentals and Tools*, IEEE Computer Society Press, Los Alamitos, CA.

Garner, B.J. & Pinnis, J. (1984) "Modelling as an Audit Technique", *The Australian Computer Journal*, Vol. 16, No. 2, pp. 48-53.

Habra, N., Le Charlier, B., Mounji, A. & Mathieu, I. (1992) "ASAX: Software Architecture and Rule-Basedd Language for Universal Audit Trail Analysis", *Proc. Computer Security - ESORICS 92*, Toulouse, France, November, pp. 435-450.

Haines, J.W., Lippmann, R.P., Fried, D.J., Tran, E., Boswell, S. & Zissman, M.A. (2001) "1999 DARPA Intrusion Detection System Evaluation: Design and Procedures", *MIT Lincoln Laboratory Technical Report*.

Hellerstein, J.L., Klein, D.A. & Milliken, K.R. (1990) *Expert Systems in Data Processing*, Addison-Wesley, Reading, MA.

Jagannathan, R., Lunt, T., Anderson, D., Dodd, C., Gilham, F., Jalali, C., Javitz, H., Neumann, P., Tamaru, A. & Valdes, A. (1993) *System Design Document: Next-Generation Intrusion Detection Expert System (NIDES)*, Technical Report, SRI International, Menlo Park, CA.

Javitz, H.S., Valdes, A., Lunt, T.F., Tamaru, A., Tyson, M. & Lowrance, J. (1993) *Next Generation Intrusion Detection Expert System (NIDES): 1. Statistical Algorithms Rationale; 2. Rationale for Proposed Resolver*, Technical Report, SRI International, Menlo Park, CA.

Krause, M. & Tipton, H.J. ed. (2001) *Information Security Management Handbook*, Fourth Edition, Auerbach, Boca Raton, FL.

Li, X. & Ye, N. (2002) "Grid- and Dummy-Cluster-Based Learning of Normal and Intrusive Clusters for Computer Intrusion Detection", *Quality and Reliability Engineering International*, Vol. 18, No. 3, pp. 231-242.

Liepins, G.E. & Vaccaro, H.S. (1989) "Anomaly Detection: Purpose and Framework", *Proc. Nat. Comp. Security Conference*, Baltimore, MD, October, pp. 495-504.

Lunt, T.F. (1993) "A Survey of Intrusion Detection Techniques", *Computers & Security*, Vol. 12, No. 4, pp. 405-418.

Lunt, T.F., Jagannathan, R., Lee, R., Listgarten, S., Edwards, D.L., Neumann, P.G., Javitz, H.S. & Valdes, A. (1988) *IDES: The Enhanced Prototype - A Real-Time Intrusion-Detection Expert System*, Technical Report, SRI-CSL-88-12, SRI International, Menlo Park, CA.

Lunt, T.F., Tamaru, A., Gilham, F., Jagannathan, R., Jalali, C., Javitz, H.S., Valdes, A. & Neumann, P.G. (1990) *A Real-Time Intrusion-Detection Expert System*, Technical Report, SRI-CSL-90-05, SRI International, Menlo Park, CA.

Luo, J & Bridges, S.M. (2000) "Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection", *International Journal of Intelligent Systems*, Vol. 15, No. 8, pp. 687-703.

Makridakis, S. & Wheelwright, S.C. (1979) "Forecasting: Framework and Overview", in *TIMS Studies in the Management Sciences*, Vol. 12, ed. Makridakis S. & Wheelwright, S.C., North-Holland, New York, pp. 1-15.

Mounji, A. & Le Charlier, B. (1997) “Continuous Assessment of a Unix Configuration: Integrating Intrusion Detection and Configuration Analysis”, in *Proceedings of the ISOC’ 97 Symposium on Network and Distributed System Security*, San Diego, California.

National Computer Security Center (1985) *Department of Defense (DoD) Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, DoD, Gaithersburg, MD.

Organisation for Economic Co-operation and Development (OECD), Ad Hoc Group of Experts (1992) *Guidelines for the Security of Information Systems [Draft]*, Directorate for Science, Technology and Industry, OECD, Paris.

Pfleeger, C.P. (1997) *Security in Computing*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ.

Proctor, P.E. & Byrnes, F.C. (2002) *The Secured Enterprise: Protecting your Information Assets*, Prentice-Hall, Upper Saddle River, NJ.

Quinlan, J.R., ed. (1989) *Applications of Expert Systems: Volume 2*, Addison-Wesley, Sydney.

Rezaee, Z., Sharbatoghlie, A., Elam, R. & McMickle, P.L. (2002) “Continuous Auditing: Building Automated Auditing Capability”, *Auditing: A Journal of Practice & Theory*, Vol. 21, No. 1, pp. 147-163.

Smaha, S.E. (1988) "Haystack: An Intrusion Detection System", *Fourth Aerospace Computer Security Applications Conference*, Orlando, Florida, December, pp. 37-44.

Snapp, S.R., Brentano, J., Dias, G.V., Goan, T.L., Heberlein, L.T., Ho, C., Levitt, K.N., Mukherjee, B., Smaha, S.E., Grance, T., Teal, D.M. & Mansur, D. (1991) "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and an Early Prototype", *Proc. 14th National Computer Security Conference*, Washington, October, pp. 167-176.

Spafford, E.H. (1989) "The Internet Worm: Crisis and Aftermath", *Commun. ACM*, Vol. 32, No. 6, pp. 678-687.

Stoll, C. (1988) "Stalking the Wily Hacker", *Commun. ACM*, Vol. 31, No. 5, pp. 484-497.

Tzafestas, S.G. ed. (1997) *Knowledge Based Systems: Advanced Concepts, Techniques and Applications*, World Scientific Publications, River Edge, NJ.

van Dijk, J.C. & Williams, P. (1990) *Expert Systems in Auditing*, Stockton Press, New York.

Winkler, J.R. & Landry, J.C. (1992) "Intrusion and Anomaly Detection: ISOA Update", *Proc. 15th National Computer Security Conference*, Baltimore, MD, October, pp. 272-281.

Witten, I.H. (1987) "Computer (In)security: Infiltrating Open Systems", *ABACUS*, Vol. 4, No. 4, pp. 7-25.

Ye, N. & Chen, Q. (2001) “An Anomaly Detection Technique based on a Chi-Square Statistic for Detecting Intrusions into Information Systems”, *Quality and Reliability Engineering International*, Vol 17, No. 2, pp. 105-112.

Ye, N., Ehiabor, T. & Zhang, Y. (2002) “First-order versus High-order Stochastic Models for Computer Intrusion Detection”, *Quality and Reliability Engineering International*, Vol. 18, No. 3, pp. 243-250.

Figure 1 Model of the MIATA System

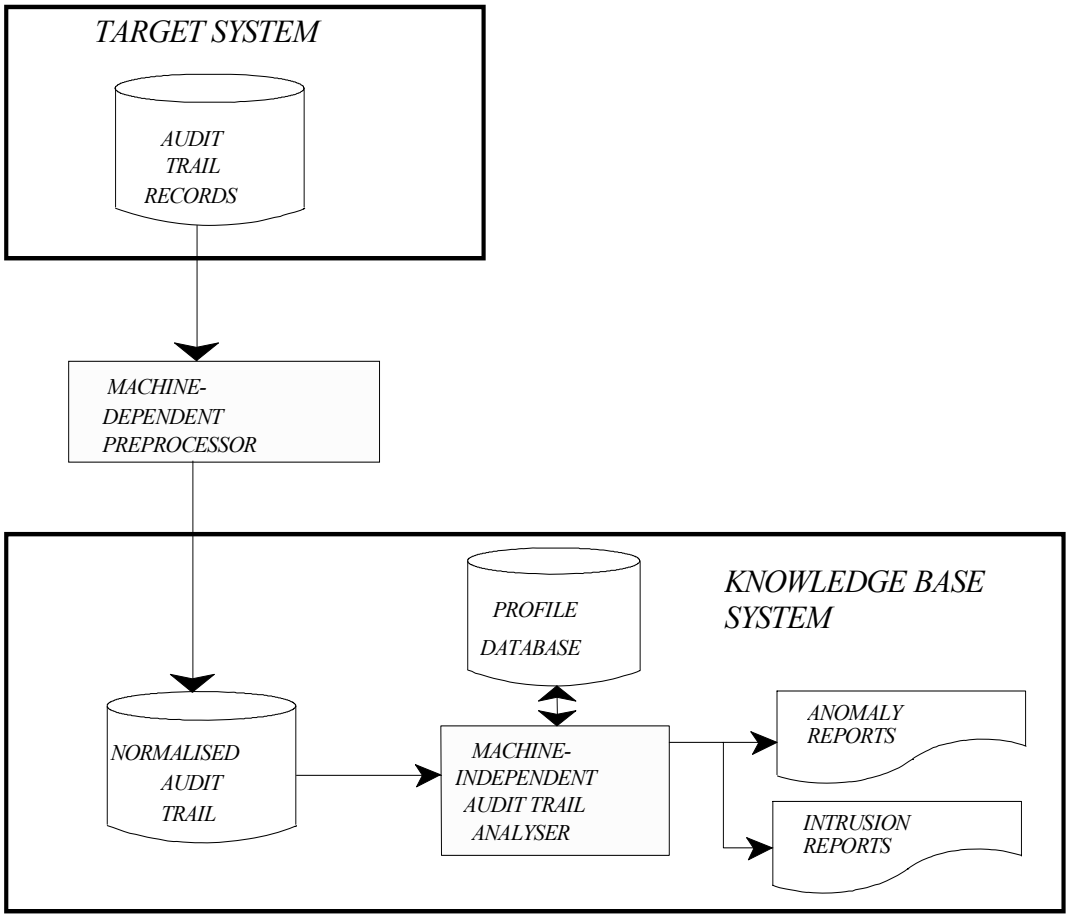

Figure 2 Mapping of Data Elements to Entities

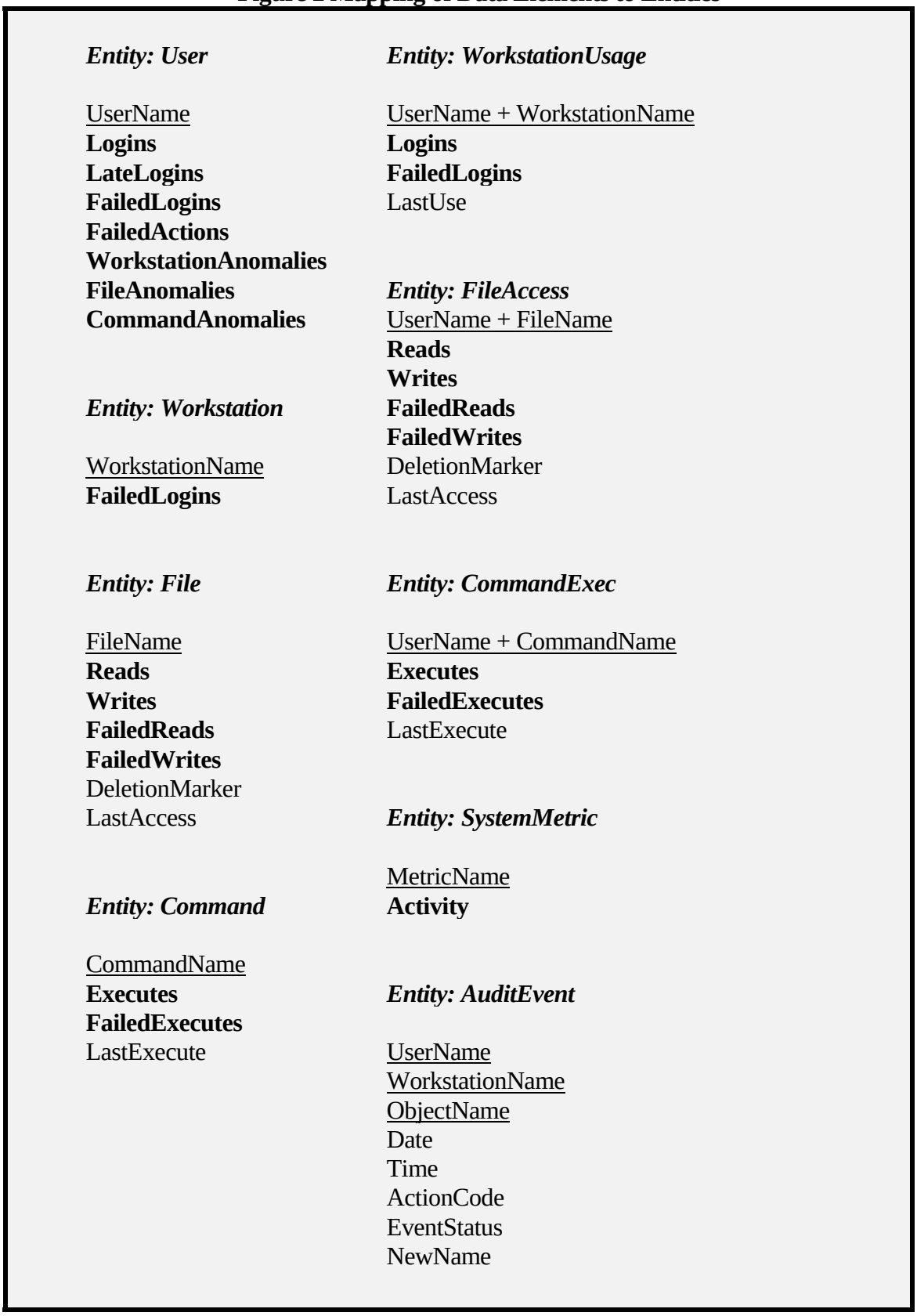

Table 1 Summary of Intruder Profiles

ANOMALOUS ACTIVITY	***PASSWORD GUESSING***	***MASQUER-ADING***	***BROWSING***
LOGINS		X	
LATE LOGINS		X	X
FAILED LOGINS	X		
FAILED ACTIONS		X	X
WORKSTATION USAGE	X	X	
FILE USAGE		X	X
FAILED FILE ACCESS		X	X
COMMAND USAGE		X	X
FAILED COMMAND USAGE		X	X