



Continuous Fraud Detection in Enterprise Systems through Audit Trail Analysis

Author

J. Best, Peter, Rikhardsson, Pall, Toleman, Mark

Published

2009

Journal Title

Journal of Digital Forensics, Security and Law

Downloaded from

<http://hdl.handle.net/10072/47590>

Link to published version

<https://www.jdfsl.org/>

Griffith Research Online

<https://research-repository.griffith.edu.au>

Continuous Fraud Detection in Enterprise Systems through Audit Trail Analysis

Peter J. Best

School of Accounting, Economics & Finance
University of Southern Queensland
Toowoomba, Queensland, 4350, Australia
Tel: (+61) 7 4631 1231 / Fax: (+61) 7 4631 5594
Email: bestp@usq.edu.au

Pall Rikhardsson

Financial Intelligence Division
Business Advisory, SAS Institute A/S
Købmagergade 7-9, DK-1150 Copenhagen K, Denmark
Tel: (+45) 7028 2506
Email: pall.rikhardsson@sdk.sas.com

Mark Toleman

School of Information Systems
University of Southern Queensland
Toowoomba, Queensland 4350 Australia
Tel: (+61) 7 4631 5593 / Fax: (+61) 7 4631 5594
Email: markt@usq.edu.au

ABSTRACT

Enterprise systems, real time recording and real time reporting pose new and significant challenges to the accounting and auditing professions. This includes developing methods and tools for continuous assurance and fraud detection. In this paper we propose a methodology for continuous fraud detection that exploits security audit logs, changes in master records and accounting audit trails in enterprise systems. The steps in this process are: (1) threat monitoring-surveillance of security audit logs for 'red flags', (2) automated extraction and analysis of data from audit trails, and (3) using forensic investigation techniques to determine whether a fraud has actually occurred. We demonstrate how mySAP, an enterprise system, can be used for audit trail analysis in detecting financial frauds; afterwards we use a case study of a suspected fraud to illustrate how to implement the methodology.

Keywords: Continuous assurance, continuous audit, fraud detection, enterprise system, accounting information systems, mySAP, audit trails.

1. INTRODUCTION

Fraud continues to be of major concern to business companies, not-for-profit organizations and governmental agencies. Recent surveys by leading accounting firms document that fraud is costing these organizations billions of dollars per year (BDO 2008; KPMG 2008; Standards Australia 2008). Furthermore, fraud means reduced macroeconomic output. Estimates indicate that fraud costs the Australian economy up to 3 billion dollars each year (Standards Australia 2008).

The incidence and financial impact of fraud seems to be steadily increasing and many organizations are ill-prepared to prevent and detect fraud (KPMG 2008). Australian Standard AS 8001-2008 (Standards Australia 2008) proposes that organizations implement a fraud detection program to quickly identify instances of fraud should preventive measures and internal controls fail. It recommends the development of systems for targeted post-transactional review and strategic use of computer systems including effective data mining and real-time transaction assessment to identify suspect fraudulent transactions. In a similar vein, PCAOB Auditing Standard No. 5 stresses the responsibility of external auditors to conduct a fraud risk assessment in planning and performing the audit of internal control over financial reporting, and to consider deficiencies in controls to prevent and detect fraud when assessing the risk of material misstatement in the financial statements (PCAOB 2007).

Few information technology innovations have had as much impact on business organizations in recent years as enterprise systems (sometimes known as enterprise resource planning systems or ERP) (Rikhardsson & Kraemmergaard 2006). Enterprise systems are off-the-shelf applications that offer a comprehensive set of functionalities supporting and integrating most business processes, including accounting, sales, purchasing and production in a single system architecture. An enterprise system has several distinctive characteristics (Norris et al. 1998):

- Multi-functional in scope – it tracks financial results (dollars), procurement (material), sales (people and goods) and manufacturing (people and resources);
- Integrated in nature, that is, when a piece of data is entered regarding one of the functions, data relevant to other functions is changed;
- Modular in structure, that is, it can be used in a way that is as expansive or narrow as an organization chooses.

Modern enterprise systems are web enabled, which can mean browser based user interfaces, standardised data exchange and web-based reporting. It has been estimated that organizations worldwide spend approximately \$18.3 billion US every year on enterprise systems (Shanks et al. 2003). These systems have significant implications for accounting and auditing in general and fraud control in particular (ITGI 2006; Bae & Ashcroft 2004, Chapman & Chua 2003;

Chapman 2005). However, enterprise systems are not typically utilised for fraud detection, certainly not in a systematic manner. These systems increase the complexity of the accounting and auditing environment but also offer new opportunities for improvements in effectiveness and efficiency of these processes (Spathis 2006).

Enterprise systems offer functionalities for continuous monitoring of controls and detecting fraudulent transactions. One such functionality is audit trails. This paper illustrates how audit trails in enterprise systems can be used for continuous fraud detection. It discusses continuous assurance and fraud detection and links these processes to enterprise systems. It explains the concept of audit trails and how they can be used for fraud detection within the context of a specific enterprise system solution – i.e. mySAP, which is described below, is a product of the German company SAP. It then proposes a methodology for continuous fraud detection that utilises various audit trails available in enterprise systems, namely security audit logs, changes in master records and accounting audit trails. This methodology is comprised of two stages: (1) threat monitoring, involving high-level surveillance of security audit logs for ‘red flags’, and (2) automated extraction and analysis of data from audit trails to document user actions. At that point, forensic investigation is used to determine whether a financial fraud has been committed.

2. DEDUCTIVE FRAUD AUDITING

The essential steps in detecting fraudulent transactions are (Albrecht et al. 2009; Institute of Internal Auditors 2003):

1. Understanding the business or operations.
2. Performing a risk analysis to identify the types of frauds that can occur.
3. Deducing the symptoms that likely frauds would generate.
4. Using computer software to search for these symptoms.
5. Investigating suspect transactions.

Each organization must incorporate within its risk management processes consideration of fraud risks. Common fraud schemes, preventive measures and symptoms (‘red flags’) are well-documented (see Albrecht et al. 2009; Baker 1999; Bologna & Lindquist 1995; Institute of Internal Auditors 2003; Koletar 2003; Zack 2003). For example, vendor frauds may involve creation of a fake vendor, purchase order, goods movement and invoice, or just a subset of these transactions. The enterprise system may pay the invoice automatically once these steps have been completed with Electronic Funds Transfer (EFT). EFT allows the transfer of money to the perpetrator’s bank account without having to establish a bank account in the name of the vendor.

The perpetrator may change the banking details for a vendor with whom the

organization transacts frequently. These details specify the bank number and the account number to be paid through bank transfer. The perpetrator switches these details to their own bank account or that of an accomplice. An invoice (often a duplicate) is entered for payment, and is subsequently paid automatically by the system (possibly without the involvement of the perpetrator). The banking details are then switched back to their original form. This is referred to as 'flipping bank details'. The respective vendor does not receive the duplicate payment and is therefore not aware of the fraud. Auditors may sample the invoice and payment, but will find them apparently genuine. Tests for duplicate invoices and payments may detect this fraud. However, many organizations have large numbers of duplicate payments, e.g. lease payments on photocopiers, and investigation of each transaction may not be feasible. This scheme is more difficult to detect if the invoice details are similar, but not identical.

Segregating vendor maintenance, invoice entry and payment can significantly reduce the risk of such frauds in the absence of collusion among personnel (Srinidhi 1994; Little and Best 2003). Weaknesses in segregation controls are common and often provide opportunities for such fraud schemes (KPMG 2008). The Sarbanes-Oxley Act (SOX) of 2002 has brought fraud and fraud detection to the fore with its emphasis on improving internal controls to reduce the risk of financial fraud. One of the important issues addressed in SOX is timely fraud detection and the link between fraud detection, internal controls and information systems (ITGI 2006). The premise is that early detection of fraud limits losses, prevents further fraud and improves controls. The real time nature of transaction data in enterprise systems and integrated accounting systems presents a specific challenge in that regard.

The next section looks at the area of continuous assurance in the context of enterprise systems.

3. CONTINUOUS ASSURANCE

Assurance services have been broadly defined as independent professional services that improve the quality of information for decision makers. In the literature, continuous assurance also appears to be a broad term for services that aim to provide continuous assurance to the buyer of these services or to a third party (Best et al. 2004; Alles et al. 2002; Elliot 2002; Rezaee et al. 2002; Sutton 2006; Jones & Xiao 2003; Yu et al. 2000; Murthy & Groomer 2004; Searcy & Woodroof 2003; Nelson 2004). The term continuous assurance is a more far-reaching term than continuous auditing as the latter service focuses on assurances only related to the annual financial report (Alles et al. 2002). Continuous assurance usually focuses on the quality of information used in internal decision making, publicly disclosed information and measures and controls for safeguarding assets (Elliot 2002; Alles et al. 2002).

To implement this process in an enterprise system environment, two main approaches have been proposed. These are the Embedded Audit Module (EAM)

approach and the Monitoring and Control Layer (MCL) approach.

The EAM approach, its benefits, drawbacks, technologies and processes have been discussed for many years (Groomer & Murthy 1989; Groomer & Murthy 2003; Alles et al. 2004; Murthy & Groomer 2004, Debreceeny et al. 2005; Alles et al. 2006). EAMs are basically independent software modules embedded in an information system where they monitor transactions and activities. Research indicates that this approach runs into practical difficulties (Debreceeny et al 2005; Kuhn & Sutton 2006). For example, concerns include having a “foreign” code in its enterprise system that is controlled by a third party – i.e. the external auditors. The maintenance of an EAM can be difficult given the changes, updates and modifications that routinely take place in enterprise systems. There are also legal liability issues should the EAM damage the host system in some way – a liability that an external auditor may be keen to avoid. Consequently, the use of EAM is limited (Debreceeny et al. 2005; Alles et al. 2006).

As an alternative to EAM, the use of a MCL has been suggested. This approach also has a rather long history in the context of information systems dating as far back as 1991 (Vasarhelyi & Halper 1991; Vasarhelyi et al. 2004; Kuhn & Sutton 2005; Alles et al. 2006; Kuhn & Sutton 2006; Du & Roohani 2007; Li et al. 2007). The MCL differs from the EAM in that it is an independent non-integrated software solution that uses middleware to extract data from the enterprise system which is to be monitored. This data can then be compared to a predefined set of rules or analysed. Currently, this approach seems more viable than the EAM approach as it does not have the same concerns regarding software maintenance, legal liability and client independence (Kuhn and Sutton 2006). Moreover, this approach is the one followed by many of the software vendors currently offering software solutions for continuous monitoring. It is also the approach explored in the section on automated continuous fraud detection later in this paper.

The monitoring activities conducted in both the EAM and the MCL approaches can focus on transaction data, which is monitored for violations of preset standards or unusual patterns. Examples could be postings on certain accounts exceeding some maximum posting limits or transaction flows exhibiting some unusual characteristics over a certain period of time. The monitoring activities may also focus on user behaviour. In most enterprise systems, users’ activities are logged. Changes in configuration, security and master records, and financial transactions are tagged with date/timestamps, user identification, and workstation identification which are collected in various audit trails. As will be discussed later, these audit trails are of different types but usually an integrated part of the system. Some audit trails must be activated before they become functional; others are a standard part of the system and are automatically present. These audit trails can then be extracted from the system and analysed for atypical user activity, authorization breaches, and profiling the activities of particular users.

In the following sections, we will discuss audit trails in enterprise systems, their

form in mySAP, and the detection of a vendor fraud using audit trail analysis.

4. AUDIT TRAILS AND ENTERPRISE SYSTEMS

Audit trails are records of user activity. They may be maintained by the operating system and by application software such as enterprise systems. Operating system audit trails record user actions, including successful and failed logins and programs executed, as well as resources consumed. Enterprise systems typically incorporate authentication processes and user roles/profiles that restrict access to the application and limit a user's capabilities to those associated with his/her job function. Potential fraud threats and related principles of segregation of duties should guide the design of user roles/profiles. Audit trails maintained by enterprise systems may include security audit logs, records of changes in master records and details of accounting transactions.

It is important to point out that these audit trails do not necessarily involve EAM nor MCL. These audit trails are part of enterprise systems and often have their own reporting facilities. However, in the context of continuous auditing they can be used for monitoring user activity. As such they can be a part of either EAM or MCL approaches.

Enterprise system **security audit logs** typically record details of each user action. These logs often include successful logins, failed logins, starting a transaction (e.g. entry of an invoice), failed attempts to start transactions (i.e. prevented by the user's role/profile), automatic locking of a user's account because of multiple failed logins, creation of new roles/profiles and changes in user master records. Configuration of the security audit log defines what events are recorded. For example, only failed activity may be recorded. These audit trails may be retained for periodic review, then archived and/or deleted.

Master records, such as those for vendors, are an important ingredient in many fraud schemes. In order for the system to distribute funds through a cheque or EFT payment, a master record must be created or modified (e.g. temporarily changing a vendor's banking details). Records of such changes in master records show user identification, type of change (e.g. create, delete, change), and contents of fields created/deleted/changed. Accounting audit trails are sets of records that permit tracing accounting transactions from their source to the updating of accounting balances, or tracing any account balance back ('drilling-down') to the relevant source transactions. They provide the organization with the ability to maintain sufficiently detailed records to answer enquiries from customers or vendors, to produce detailed reports and monthly statements for customers, and to provide data for managerial decision-making. Master record changes and accounting audit trails are retained on-line usually for the entire fiscal year, and archived for several years to satisfy the requirements of taxation and company legislation.

The audit trails of enterprise systems can serve several purposes:

1. Review of access: Audit trails allow examination of the history of access by individual users or groups of users, showing actions performed or attempted. Audit trails also can report which users have performed specific functions, such as changes to vendor master records or the entry of vendor invoices. Analysis of audit trails may also reveal limitations in the organization's security model and its implementation.
2. Review of changes in security: Changes made to the security of the system can be reviewed periodically by an independent person for authorisation and integrity.
3. Review of attempts to by-pass security: Audit trails may be reviewed for attempts and repeated attempts by users and intruders to perform unauthorised functions.
4. Deterrent against attempts to by-pass security: Users should be aware of the existence of audit trails and their routine review as a deterrent against attempts to by-pass security.
5. Fraud detection: Audit trails can be used to detect potential fraud by searching for red flags. Fraudulent activity may be perpetrated by real users acting in their own name, by users acting in collusion with other users, by real users masquerading as other users, or by intruders masquerading as authorised users. In each case, the actions of these 'users' are recorded in audit trails and these can be scrutinised for activities that are recognised as red flags for particular types of fraud.

The next section examines how these types of audit trails are implemented in mySAP.

5. MYSAP AND SYSTEM SECURITY

The mySAP solution combines complete and scalable software for enterprise resource planning with a flexible, open technology platform (the SAP NetWeaver) that can leverage and integrate SAP and non-SAP systems. It builds on and extends functionalities in earlier SAP solutions (SAP R/2 and SAP R/3), which have been on the market since the 1970s. SAP offers integrated modules for accounting, production planning, materials management, sales and distribution, quality management, project management and more. mySAP allows complex enabling companies to integrate most financial, people, asset and data management tasks in one comprehensive IT infrastructure. The mySAP framework includes four individual solutions: (1) mySAP Financials, (2) mySAP Human Capital Management, (3) mySAP Operations and (4) mySAP Corporate Services.

The system provides functionality supporting internal control assessment, such as reporting on changes in user profiles and segregation of duties. End-users and 'system' users access the system through the same authentication process

requiring the entry of a client identification, user name, password and language. These users share the same main menu to access accounting, logistics (procurement, sales, production) and human resource transactions, as well as the mySAP program development, security administration and configuration functions. Accordingly, access controls must be implemented to restrict the actions of all users in conformance with their assigned roles.

Access and user controls are implemented in mySAP using roles, profiles and authorizations which are assigned to users. The individual functions (menu options) are identified within the system using transaction codes. For example, the function to change vendor master records has the transaction code FK02. Entry of a vendor invoice is FB60. Associated with each transaction code is a set of authorizations which must be assigned to a given user to allow them to perform that function. User profiles consisting of sets of authorizations and other profiles should be designed according to principles of segregation of incompatible transaction codes in order to reduce opportunities for fraud (Little & Best 2003). Any user who has the authority to change a vendor's banking details and enter a vendor invoice has the opportunity to commit fraud.

Security administrators use mySAP's profile generator software to design generic roles which may be assigned to individuals. To illustrate, a role may be designed for vendor maintenance officers, consisting of just the transaction codes required for that role and considering relevant segregation of duties principles. Such a role should not include the transaction code FB60 Enter Vendor Invoice. Profile generator automates the process of building profiles with the required authorizations for roles. Given the large number of transaction codes in the system (at least 125,000) and some uncertainty regarding appropriate segregation principles, some users may be assigned authorizations which permit certain fraud schemes. Accordingly, there is a need for auditing of access controls and automated approaches for fraud detection which analyse audit trails.

Auditors typically plan to evaluate and test the client's security model for compliance. This model consists of a set of roles (or profiles) and their assignment to users. The transactions (and authorizations) assigned to each role are also documented. The security model is 'desk-checked' for completeness and proper segregation of duties, and then tested for proper implementation on a 'sample' basis by interrogating authorizations, profiles, roles and user master records. Proper segregation of organizational responsibilities is a critical concern in this process.

Authorizations may also be audited by interrogating system security tables to identify authorizations assigned to users and the corresponding transaction codes which may be executed. This may be accomplished using software developed in-house or acquired from third-party providers, or using standard mySAP reports.

6. DETECTING VENDOR FRAUD WITH AUDIT TRAIL ANALYSIS IN MYSAP

mySAP offers managers and auditors increased facilities for monitoring user activities in the system, including potential fraudulent activities. These activities are collected automatically in mySAP's audit trails. Below we describe these facilities and illustrate how a vendor fraud based on changes in vendor banking details and duplicate invoice entry may be detected through audit trail analysis.

The security audit log facility provides a high-level overview of user activity at the transaction code level. A profile is created and filters are defined specifying which events are recorded in the log (transaction SM19). Selected events are stored in a daily audit file on each application server. These audit files are retained until deleted.

Filters specify which clients and users are to be monitored. Events may be selected for logging according to audit class, such as logons, transaction starts, and user master changes, or according to event class - critical events, critical events combined with important events, or all events. Alternatively, a set of individually selected events may be selected as a detailed audit configuration. Once the filter(s) and profile are activated, the application server must be restarted and then logging commences.

Table 1 illustrates the relationship between audit classes, event classes and the message text for individual events.

Audit records contain the following fields for each logged event: Date, Time, Client, User-id, Transaction Code, Terminal Name (computer name from Windows), Message Identifier, and Message Text. A reporting facility is provided for the security audit log. Reports may be produced for specified date ranges, users, transaction codes, audit classes, event classes and messages.

Audit Class	Event Class	Message Text
Dialog Logon	Non-Critical	User Logoff
	Important	Logon Successful (Type = \$A)
	Important	Logon Failed (Reason = \$B, Type = \$A)
	Critical	Logon Failed (Reason = \$B, Type = \$A)
	Critical	User &B Locked in Client \$A after Erroneous Password Attempts
	Critical	User &B in Client &A Unlocked After Being Locked Due to Invalid Password Entered
Transaction Start	Non-Critical	Transaction &A Started
	Critical	Start Transaction &A Failed
User Master Change	Non-Critical	Password changed for user &B in client &A
	Important	User &A Deleted
	Important	User &A Locked
	Important	User &A Unlocked
	Important	Authorizations for User &A Changed
	Important	User Master Record &A Changed
	Important	Authorization/Authorization Profile &B Created
	Important	Authorization/Authorization Profile &B Deleted
	Important	Authorization/Authorization Profile &B Changed
	Critical	User &A Created
	Critical	Authorization/Authorization Profile &B Activated
Other Events	Important	Download &A Bytes to File &C
	Important	Digital Signature (Reason = &A, ID = &B)
	Critical	Digital Signature Error (Reason = &A, ID = &B)
	Critical	Password check failed for user &B in client &A
System	Critical	Audit Configuration Changed
	Critical	Application Server Stopped
	Critical	Application Server Started
	Critical	Audit Slot &A Inactive
	Critical	Audit Active Status Set to &1

Table 1: Security Audit Log – Examples of events that can be logged

These functionalities can be used to detect fraudulent user behaviour. Figure 1 presents an excerpt from the security audit log showing a range of logged events. Of particular note are the following:

- User HACKERW uses workstation 1 in room S826 (see column 6 – Terminal).
- On 01.04.2008, HACKERW attempted to run transaction F110 (column 5 – Transaction Code) Vendor Payments unsuccessfully. Message-id AU4 signifies a failed action.
- HACKERW performed changes to vendor master records using transaction FK02 on 03.04.2008 and 05.04.2008.
- User SMITHY apparently had 3 failed logons on 08.04.2008 from the same workstation as used by HACKERW. The user was automatically locked and had to be unlocked by a security administrator ZADMIN01.

- On 24.04.2008, ZADMIN01 used transaction SU01 to create user ZMYUSER. Authorizations were assigned to this user.
- On the same day at 11:31:25, ZADMIN01 used transaction PFCG Profile Generator to create a new role Z:VENDM50 (which is assigned a series of transaction codes).
- Transaction SUPC (Generate Profiles) was then used to generate the authorizations and profile for the new role.
- ZADMIN01 then proceeded to assign the new role to user ZMYUSER.
- User ZMYUSER then apparently logged on to client 600 and was required to change the initial password.
- ZMYUSER used transaction FK02 to perform vendor maintenance and then logged off.
- User ZADMIN01 used transaction SU01 to delete user ZMYUSER (This can be done even after this user has performed activity in the system).

Changes in master records are stored in two tables – CDHDR Change Document Headers and CDPOS Change Document Items. Changes include creation and deletion of master records and changes in fields. Each change document header record in table CDHDR specifies: Client, Object class of the master record, e.g. category of vendor, customer, general ledger account, cost centre, etc., Object value, i.e. vendor number, cost centre code, Change document number, User name who made the change, Date, Time, and Transaction code, e.g. FK02 Change Vendor Master Record.

For each change document number, there are corresponding change document items in the CDPOS table. Change document items have the following fields: Client, Object class of the master record, e.g. category of vendor, customer, general ledger account, cost centre, etc., Object value, i.e. vendor number, cost centre code, Change document number, Table name, e.g. LFBK – Vendor Master (Bank Details), Table record key, Field name, Change type - U(pdate), I(nsert), E(delete single field), D(elete).

Figure 2 illustrates how changes in banking details for vendor 100163 would appear in these tables. The original bank recording number and account number is 123-456 1234567. This vendor was created by user SMITHY on 15.02.2008 using transaction code FK01 (See first row in Table CDHDR and first row in Table CDPOS). On 03.04.2008, these details were changed by user HACKERW to 123-456 7777777 (see second row in Table CDHDR and second row in Table CDPOS), and then restored to the original values on 05.04.2008 (see the third row in each of the Tables).

Date	Time	Client	User	Trans Code	Terminal	Message Id.	Message Text
01.04.2008	08:55:04	600	HACKERW		S826-01	AU2	Logon Failed (Reason = 1, Type = A)
01.04.2008	08:56:30	600	HACKERW		S826-01	AU1	Logon Successful (Type=A)
01.04.2008	11:25:09	600	HACKERW		S826-01	BU2	Password changed for user HACKERW
01.04.2008	12:31:54	600	HACKERW	FK01	S826-01	AU3	Transaction FK01 Started
01.04.2008	13:43:11	600	HACKERW	F110	S826-01	AU4	Start Transaction F110 Failed
01.04.2008	18:18:12	600	HACKERW		S826-01	AUC	User Logoff
03.04.2008	08:37:40	600	HACKERW			AU1	Logon Successful (Type=A)
03.04.2008	10:20:25	600	HACKERW	FK02	S826-01	AU3	Transaction FK02 Started
03.04.2008	10:23:44	600	HACKERW	FB60	S826-01	AU3	Transaction FB60 Started
05.04.2008	17:14:31	600	HACKERW	FK02	S826-01	AU3	Transaction FK02 Started
08.04.2008	08:55:04	600	SMITHY		S826-01	AU2	Logon Failed (Reason = 1, Type = A)
08.04.2008	08:55:06	600	SMITHY		S826-01	AU2	Logon Failed (Reason = 1, Type = A)
08.04.2008	08:55:08	600	SMITHY		S826-01	AU2	Logon Failed (Reason = 1, Type = A)
08.04.2008	08:55:09	600	SMITHY			AUM	User SMITHY Locked in Client 600 After Erroneous Password Checks
08.04.2008	09:05:01	600	ZADMIN01	SU01	B315-01	AU3	Transaction SU01 Started
08.04.2008	09:05:02	600	ZADMIN01		B315-01	AUN	User SMITHY in Client 600 Unlocked After Being Locked Due to Inval. Password Entered
24.04.2008	11:15:33	600	ZADMIN01		B315-01	AU1	Logon Successful (Type=A)
24.04.2008	11:16:16	600	ZADMIN01	SU01	B315-01	AU3	Transaction SU01 Started
24.04.2008	11:18:38	600	ZADMIN01	SU01	B315-01	AU7	User ZMYUSER Created
24.04.2008	11:18:39	600	ZADMIN01	SU01	B315-01	AUB	Authorizations for User ZMYUSER Changed
24.04.2008	11:28:34	600	ZADMIN01	SU03	B315-01	AU3	Transaction SU03 Started
24.04.2008	11:31:09	600	ZADMIN01	SU03	B315-01	AUU	Authorization Z:AUTH5001/F_KNA1_BUK Activated
24.04.2008	11:31:25	600	ZADMIN01	PFCG	B315-01	AU3	Transaction PFCG Started
24.04.2008	11:33:05	600	ZADMIN01	SUPC	B315-01	AU3	Transaction SUPC Started
24.04.2008	11:36:23	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_BKPF_BEK Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_BKPF_BLA Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_BKPF_BUK Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_BKPF_GSB Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_BKPF_KOA Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_AEN Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_APP Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_BEK Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_BUK Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_GEN Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/F_LFA1_GRP Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Authorization Z:VENDM50_00/S_TCODE Activated
24.04.2008	11:36:24	600	ZADMIN01		B315-01	AUU	Profile Z:VENDM50_Activated
24.04.2008	11:37:15	600	ZADMIN01	SU01	B315-01	AU3	Transaction SU01 Started
24.04.2008	11:37:47	600	ZADMIN01	SU01	B315-01	AUD	User Master Record ZMYUSER Changed
24.04.2008	11:37:48	600	ZADMIN01	SU01	B315-01	AUB	Authorizations for User ZMYUSER Changed
24.04.2008	11:38:10	600	ZMYUSER		B315-01	AU1	Logon Successful (Type=A)
24.04.2008	11:38:18	600	ZMYUSER		B315-01	BU2	Password changed for user ZMYUSER in client 600
24.04.2008	11:39:00	600	ZMYUSER	FK02	B315-01	AU3	Transaction FK02 Started
24.04.2008	11:40:07	600	ZMYUSER		B315-01	AUC	User Logoff
24.04.2008	11:56:16	600	ZADMIN01	SU01	B315-01	AU3	Transaction SU01 Started
24.04.2008	11:58:38	600	ZADMIN01	SU01	B315-01	AU8	User ZMYUSER Deleted
24.04.2008	18:18:12	600	ZADMIN01		B315-01	AUC	User Logoff

Figure 1 - mySAP Security Audit Log

TABLE CDHDR – Change Document Headers								
Client	Object Class	Object Id	Change No.	User	Date	Time	Transaction Code	
600	KRED	0000100163	0000446154	SMITHY	15.02.2008	15:10:39	FK01	
600	KRED	0000100163	0000446258	HACKERW	03.04.2008	10:21:05	FK02	
600	KRED	0000100163	0000446351	HACKERW	05.04.2008	17:15:39	FK02	

TABLE CDPOS – Change Document Items									
Client	Object Class	Object Id	Change No.	Table	Table Key		Field	Change	
600	KRED	0000100163	0000446154	LFBK	6000000100163AU	123-456	1234567	KEY	I
600	KRED	0000100163	0000446258	LFBK	6000000100163AU	123-456	7777777	KEY	I
600	KRED	0000100163	0000446351	LFBK	6000000100163AU	123-456	1234567	KEY	I

Table Key = Client, Vendor No., Bank Country, Bank Key, Bank Acct No.

Figure 2 - Changes in Vendors Banking Details in mySAP

Figure 3 provides an overview of tables storing mySAP financial accounting audit trails. In following the audit trail from Figure 2 to Figure 3, it can be seen that HACKERW used transaction code FB60 (see table BKPF) to enter a vendor invoice on 03.04.2008. This transaction was recorded as document number 1000000201 in table BKPF Accounting Document Headers. The user name, date and transaction code are stored in this record. There are three debit/credit entries corresponding to this document in table BSEG Accounting Document Line Items. Every posting to a general ledger reconciliation (control) account also specifies the relevant subsidiary ledger record. Since account number 209000 (Table SKAT) is the Accounts Payable account, the vendor number (100163) is also recorded in the line item record (Table LFA1). Tables BKPF and BSEG store the posting history for both general ledger accounts and subsidiary ledger records, thereby facilitating both integration of data and automatic reconciliation of subsidiary ledgers with reconciliation accounts. General ledger account texts (names) are stored in table SKAT. Vendor general data including vendor name, date created and creating user are stored in table LFA1.

As can be seen in the above, the data describing the fraud is well-documented in the audit trails in the enterprise system. However, detecting user activities and analysing them for fraud potential is a laborious task if done manually. We propose a methodology based on automated continuous analysis of audit trails.

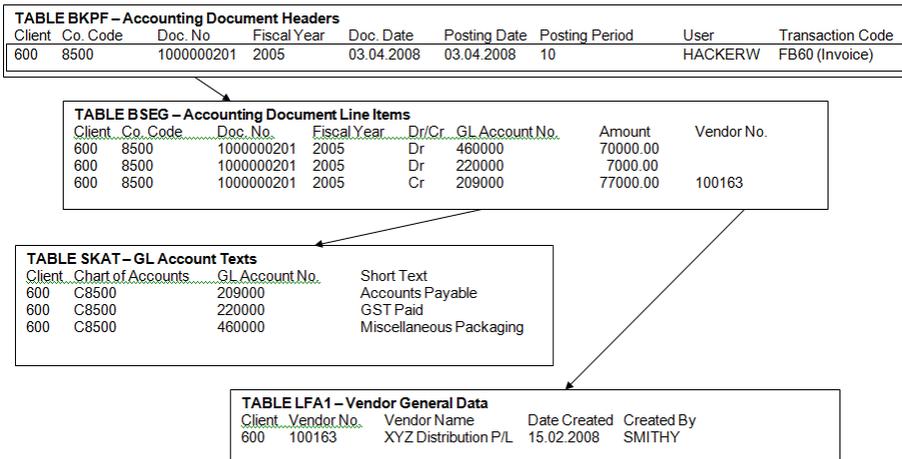


Figure 3 - mySAP Audit Trail

7. AUTOMATED CONTINUOUS FRAUD DETECTION METHODOLOGY

Using mySAP as an example, we propose an MCL-based methodology for fraud detection that utilises the security audit logs, changes in master records and accounting audit trails present in mySAP. This methodology is comprised of two stages: (1) threat monitoring, which involves high-level surveillance of security audit logs for ‘red flags’, and (2) automated extraction and analysis of data from audit trails to provide documentation of user actions. These two stages are demonstrated for the vendor fraud scenario.

Stage 1 involves threat monitoring (routine scanning) of security audit logs. These logs should be extracted for regular review and retained to provide a permanent actual user profile for each user. The organization may develop a database application storing security audit logs for the past year, and user profiles (the set of transaction codes performed by each user during specific time periods, e.g. last week, last month). Queries should be available to list users who have performed specified transaction codes. Standard reports should be available to present any specified user’s profile and to highlight changes in users’ profiles over time. A knowledge base system may also be developed to generate forecasts of expected user activity. Changes in actual user behaviour may then be detected promptly and investigated (Best et al. 2004).

To detect specific fraud threats, a standard report should present a list of users and log details where a critical combination of transaction codes has been performed by a user. For example, any user who has performed vendor master record changes (transaction code FK02) and vendor invoice entry (FB60) should be classified as suspicious, since the combination of these functions may signal the

flipping of bank details and vendor fraud as described in the diction on deductive fraud auditing. A table of suspects should be generated to facilitate detailed analyses of master record changes and accounting transactions. In Figure 1, HACKERW, who executed these transactions, would be identified as a potential suspect. Identification of the affected vendors requires data extraction from the appropriate audit trails.

Stage 2 requires routine extraction of master record changes and accounting audit trails, as a foundation for further analysis of suspect behaviour for the set of chosen fraud schemes. The following data may be extracted from mySAP through the data dictionary or using remote function calls.

1. Change document headers: Records are extracted from table CDHDR (see Figure 2) for changes involving vendor account groups, the current fiscal year and critical transaction codes (e.g. FK02).
2. Change document items: Records are extracted from table CDPOS (see Figure 2) for INSERT (I) changes involving vendor account groups, table LFBK, and field KEY.
3. Accounting document headers: Records are extracted from table BKPF (see Figure 3) for documents involving the target company code, current fiscal year, and transaction codes associated with fraud schemes (e.g. FB60 – vendor invoice entry, F110 – vendor payment).
4. Accounting document line items: Records are extracted from table BSEG (see Figure 3) for postings (rows) involving the target company code, current fiscal year, and accounts payable general ledger accounts.

Change document headers and change document items may be used to produce a detailed analysis of the banking details changes performed by the suspect users. In particular, the relevant vendor numbers are identified. For example, examining the data in Figure 2, it is evident that HACKERW has changed the banking details for vendor 100163, on 03.04.2008 and switched them back on 05.04.2008. The accounting document headers and line items may be used to present the accounting transactions entered by the suspects and invoice and payment transactions for the associated vendors. The invoice (FB60) posted by HACKERW on 03.04.2008 was for \$77,000 (including sales tax) to vendor 100163. Such an analysis may be correlated to test for specified sequences of events such as: changed vendor details, entered invoice, payment of invoice, changed vendor details. If payment occurs before 05.04.2008, it appears that HACKERW may have successfully perpetrated a vendor fraud since payment is made to the changed banking details before they are flipped back. A thorough investigation is still required to determine whether this is the case.

8. CASE STUDY RESULTS – A LARGE AUSTRALIAN COMPANY

The application of this methodology assisted in a fraud investigation for a large Australian company with a very large mySAP system implementation.

Basic application of threat monitoring of the security audit log revealed that a terminated system administration person (SADMIN01) had since logged in and changed a password and the profile of his spouse, also in system administration (SADMIN02). A high risk of unauthorised activity and/or fraud was identified, possibly involving SADMIN01, SADMIN02 or both users working in collusion. The findings from the application of this methodology are summarised below.

More thorough threat monitoring was instituted covering a period of over four years. It seemed that both SADMIN01 and SADMIN02 had been engaged in vendor maintenance (FK02) and invoice entry (FB60) activity. However, other system administrators had also performed similar functions, in some cases to a much larger extent. Concern was raised that members of the SADMIN group could be working in collusion with SADMIN01 and/or SADMIN02. These users were also responsible for user security, including the creation and maintenance of user master records and profiles. There was also an increased risk of fake users in the system, engaged in fraudulent activities.

SADMIN01, SADMIN02, SADMIN04, SADMIN06 and SADMIN11 had made changes to vendors (FK02) as follows:

- SADMIN01 – 2 changes to only 2 vendors. No flipping of bank details was feasible.
- SADMIN02 – 689 changes, with more than 1 change to only 13 vendors. Flipping was feasible. No changes were made to vendors maintained by SADMIN01.
- SADMIN04 – 7 changes with more than 1 change to only 2 vendors. Flipping was feasible. No changes were made to vendors maintained by SADMIN01.
- SADMIN06 – 2585 changes with more than 1 change to more than 500 vendors. No changes were made to vendors maintained by SADMIN01.
- SADMIN11 – 4403 changes with more than 1 change to more than 400 vendors. 1 change was made to a vendor maintained by ZADMIN01.

The vendors maintained by SADMIN01 and SADMIN02 were targeted to investigate the presence of flipping activity. Numerous changes to banking details of vendors were performed by SADMIN11 on Christmas Eve in year 1, which were subsequently changed (back in some cases) by SADMIN02 after the Christmas/New Year break. The apparent flipping of bank details occurred for large numbers of vendors, but these details remained in force for several weeks. A

small number of immaterial financial transactions were entered for these vendors during this period by SADMIN04 and SADMIN06. There was no evidence of exploiting the changed bank details during this period to commit material fraud. Internal audit were charged with the task of investigating this unusual set of events.

Flipping of bank details could be indicated by the apparent sharing of bank accounts. This occurs when an invoice is paid to the account of a vendor, which has the same bank details as another vendor in the system. Some evidence of bank account sharing by vendors was revealed, but these cases involved spouses or multiple vendor master records for the same vendor. These were examined and were considered genuine.

SADMIN users were also engaged in the entry of financial transactions, including FB60. An examination was performed on the financial transactions entered by SADMIN01 and SADMIN02 for the vendors changed by these users. These postings were trivial in amount. Only five were payments. Financial transactions for these vendors entered by other SADMIN users appeared normal and did not involve the redirection of payments to other bank accounts.

Despite the alert raised on discovery of the abnormal activity by SADMIN01 (and SADMIN02), there was no evidence found of material fraud by that user. It seemed that SADMIN users performed the functions of normal users – maintaining vendors, entering invoices and paying vendors. There were breaches in the normal segregation of duties principles: (1) separating the functions performed by accounting users from those of system administrators; and (2) separating vendor maintenance, entry of invoices/postings and payment functions. The financial transactions entered by SADMIN01 and SADMIN02 appeared trivial vendor changes.

This investigation led to wide changes to user profiles in this company. Segregation amongst normal users seemed to be following appropriate segregation principles. However, vendor maintenance and invoice entry were not adequately segregated. Two accounts payable personnel were subsequently assigned new profiles for vendor maintenance and invoice entry respectively. It was determined that SADMIN users had been able to perform vendor maintenance, invoice entry and payment transactions because of their assigned user profiles. These were mainly SAP_ALL profiles which give the user unlimited access to system functions. It was necessary to design new profiles for SADMIN users that explicitly provided authorizations for their roles in system administration. This had the effect of removing their ability to perform the functions of accounting users.

This investigation highlights the potential vulnerability to vendor fraud that may arise from inadequate segregation of duties and the need for automated continuous fraud detection solutions.

9. LIMITATIONS

It is important to acknowledge the limitations of the fraud detection methodology presented in this paper. The audit trails that are maintained by enterprise systems are the basis for this methodology. As such, their integrity is paramount in assessing the usefulness of this methodology for detecting actual fraud.

The behaviour of individual users will be recorded in detail in the audit trails. However, system administrators, often called ‘super-users’ given their unlimited privileges, may be able to selectively edit audit trail data, such as entries in the security audit log, to remove evidence of ‘red flags’ associated with their own activity. Similarly, intruders in the system who are masquerading as authentic users may target these super-users and exploit these capabilities to remove any trace of their activities in the system.

Accordingly, it is acknowledged that the fraud detection methodology proposed in this paper may not be useful in detecting fraud by super-users, nor intruders who masquerade as these powerful users. However, this methodology is very useful in detecting fraudulent behaviour by normal users or intruders masquerading as such users, who lack these capabilities. Most reported cases of fraud seem to be perpetrated by such unsophisticated users.

10. CONCLUSION

This paper has addressed some of the challenges enterprise systems and continuous assurance pose to the accounting and auditing professions. One important challenge is how fraud detection can be integrated into continuous assurance services in the enterprise system. This paper has demonstrated one possible method for continuous fraud detection in enterprise systems based on extraction of data from audit trails. It proposes a methodology using audit trail analysis where user behaviour is monitored and analysed to detect specific fraud scenarios. Its application was demonstrated using the mySAP solution. The application of this methodology in investigating potential material fraud was also demonstrated using a case study of an Australian company.

Looking at the enterprise systems market and current vendor strategies, developments could be expected to take one of two routes. One is that continuous assurance tools continue to be stand-alone applications that extract data from the enterprise system – i.e. the MCL approach. The other is that enterprise system vendors will incorporate these systems in their enterprise systems solutions and develop EAM for use by auditors.

Accounting information systems have undergone considerable change over the past decade, and more extensive changes are likely to come in the future. Assurance services and associated technologies must keep pace with these changes. Accordingly, the development of continuous monitoring tools and fraud detection will be rich research areas. This includes further research into the applicability of EAM and MCL approaches respectively, the differences

and similarities between different enterprise system vendor approaches, the practices of auditing firms regarding continuous auditing and determinants of market demand for continuous assurance and continuous assurance tools.

11. REFERENCES

- Albrecht, W.S., Albrecht, C.C., Albrecht, C.D. and Zimbelman, M. (2009), *Fraud Examination, Third Edition*, Thomson/South-Western, Mason OH.
- Alles, M., Kogan, A. and Vasarhelyi, M.A. (2002), "Feasibility and Economics of Continuous Assurance", *Auditing*, 21(1): 126-138.
- Alles, M., Kogan, A. and Vasarhelyi, M.A. (2004), "Restoring Auditor Credibility: Tertiary Monitoring and Logging of Continuous Assurance Systems", *International Journal of Accounting Information Systems*, 5: 183-202.
- Alles, M., Brennan, G., Kogan, A. and Vasarhelyi, M.A. (2006), "Continuous Monitoring of Business Process Controls: A Pilot Implementation of a Continuous Auditing System at Siemens", *International Journal of Accounting Information Systems*, 7: 137-161.
- Bae, B. and Ashcroft, P. (2004), "Implementation of ERP systems: Accounting and Auditing Implications", *Information System Control Journal*, 5: 43-56.
- Baker, K. (1999), *Internal Control and Fraud Prevention in Hospitality Operations*, Pearson Education - Hospitality Press, Sydney.
- BDO (2008), 'BDO Not-For-Profit Fraud Survey', www.bdo.com.au, 15/1/2009.
- Best, P.J., Mohay, G. and Anderson, A. (2004), "Machine-Independent Audit Trail Analysis – A Decision Support Tool for Continuous Audit Assurance", *International Journal of Intelligent Systems in Accounting, Finance and Management*, 12: 85-102.
- Bologna, J.G. and Lindquist, R.L. (1995), *Fraud Auditing and Forensic Accounting: New Tools and Techniques*, Wiley, New York.
- Chapman, C., and Chua, W.F. (2003), 'Technology-driven integration, automation and standardisation of business process: Implications for accounting', in *Management Accounting in the Digital Economy*, ed. Bhimani, A., Oxford University Press, Oxford, 74-79.
- Chapman, C. (2005), "Not Because They are New. Developing the Contribution of Enterprise Resource Planning Systems to Management Control Research", *Accounting, Organizations and Society*, 30: 685–689.
- Debreceeny, R.S., Gray, G.L., Jun-Jin Ng, J., Siow-Ping Lee, K. and Yau, W. (2005), "Embedded Audit Modules in Enterprise Resource Planning Systems: Implementation and Functionality", *Journal of Information Systems*, 19(2): 7-

27.

Du, H. and Roohani, S. (2007), "Meeting Challenges and Expectations of Continuous Auditing in the Context of Independent Audits of Financial Statements", *International Journal of Auditing*, 11(2): 133-146.

Elliot, R.K. (2002), "Twenty-First Century Assurance", *Auditing*, 21(1): 139-146.

Groomer, S.M., and Murthy, U.S. (1989), "Continuous Auditing of Database Applications: An Embedded Audit Module Approach", *Journal of Information Systems*, 3(2): 53-69.

Groomer, S.M. and Murthy, U.S. (2003), "Monitoring High Volume On-line Transaction Processing Systems Using a Continuous Sampling Approach", *International Journal of Auditing*, 7: 3-19.

Institute of Internal Auditors (2003), *Proactively Detecting Occupational Fraud Using Computer Audit Reports*, Institute of Internal Auditors Research Foundation, Altamonte Springs, Florida.

ITGI – IT Governance Institute (2006), *IT Control Objectives for Sarbanes-Oxley*, Second Edition.

IT Governance Institute, Rolling Meadows IL, www.isaca.org, 21/4/2008.

Jones, J.M. and Xiao, J.Z. (2003), "Internet Reporting: Current Trends and Trends by 2010", *Accounting Forum*, 27(2): 132-165.

Koletar, J.W. (2003), *Fraud Exposed: What You Don't Know Could Cost Your Company Millions*, Wiley, New York.

KPMG (2008), 'KPMG 2008 Fraud Survey', www.kpmg.com.au, 20/1/2009.

Kuhn, J.R. and Sutton, S. (2005), 'Learning from WorldCom: Implications for Fraud Detection Through Continuous Assurance', 10th World Continuous Auditing and Reporting Symposium, November, Newark, NJ.

Kuhn, R. and Sutton, S. (2006), Commentary On "Embedded Audit Modules In Enterprise Resource Planning Systems: Implementation And Functionality", Working Paper, Kenneth G. Dixon School of Accounting, University of Central Florida.

Li, Y., Roget, J.N., Rydl, L. and Hughes, J. (2007), "Achieving Sarbanes-Oxley Compliance with XBRL-Based ERP and Continuous Auditing", *Issues in Information Systems*, 8(2): 430-436.

Little, A.G. and Best, P.J. (2003), "A Framework for Segregation of Duties in an SAP R/3 Environment", *Managerial Auditing Journal*, 13(5): 419-430.

Nelson, L. (2004), "Stepping into Continuous Audit", *Internal Auditor*, April, 27-29.

- Norris, G., Wright, I., Hurley, J.R., Dunleavy, J. and Gibson, A. (1998), *SAP: An Executive's Comprehensive Guide*, Wiley, New York.
- Murthy, U. and Groomer, S.M. (2004), "A Continuous Auditing Web Service Model for XML-Based Accounting Systems", *International Journal of Accounting Information Systems*, 5: 138-163.
- PCAOB (2007), *Auditing Standard No. 5 An Audit of Internal Control Over Financial Reporting that is Integrated with an Audit of Financial Statements*.
- Rezaee, A., Sarbatoghlie, A., Elam, R. and McMickle, P.L. (2002), "Continuous Auditing: Building Automated Auditing Capability", *Auditing*, 21(1): 145-163.
- Rikhardsson, P.M. and Kraemmergaard, P. (2006), "Identifying the Impacts of Enterprise System Implementation and Use: Examples from Denmark", *International Journal of Accounting Information Systems*, 7(1): 36-49.
- Searcy, D. and Woodroof, J.B. (2003), "Continuous Auditing: Leveraging Technology", *The CPA Journal*, May: 46-48.
- Shanks, G., Seddon, P.B. and Willcocks, L.P. (2003), 'ERP – The Quiet Revolution?' in *Second-Wave Enterprise Resource Planning Systems: Implementing for Effectiveness*, eds. Shanks, G., Seddon, P.B. and Willcocks, L.P., Cambridge University Press, Cambridge, 1-22.
- Srinidhi, B. (1994), "The Influence of Segregation of Duties on Internal Control Judgements", *Journal of Accounting, Auditing & Finance*, 9(3): 423-444.
- Spathis, C. (2006), "Enterprise Systems Implementation and Accounting Benefits", *Journal of Enterprise Information Management*, 19(1): 67-82.
- Standards Australia (2008), 'Australian Standard AS 8001-2008 - Fraud and Corruption Control', www.saiglobal.com/shop/Script/search.asp, 10/2/2009.
- Sutton, S. (2006), "Extended-Enterprise System Impact on Enterprise Risk Management", *International Journal of Accounting Information Systems*, 19(1): 97-114.
- Vasarhelyi, M.A., Alles, M. and Kogan, A. (2004), "Principles of Analytic Monitoring for Continuous Assurance", *Journal of Emerging Technologies in Accounting*, 1: 1-21.
- Vasarhelyi, M.A. and Halper, F.B. (1991), "The Continuous Audit of Online Systems", *Auditing: A Journal of Practice and Theory*, 10(1):110-125.
- Yu, C.-C., Yu, H.-C. and Chou, C.-C. (2000), "The Impact of Electronic Commerce on Auditing Practices: An Auditing Process Model for Evidence Collection and Validation", *International Journal of Intelligent Systems in Accounting, Finance & Management*, 9: 195-216.

Zack, G.M. (2003), *Fraud and Abuse in Non-Profit Organizations*, Wiley, New York.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.