

A State of the Art Survey of Fraud Detection Technology

Ulrich Flegel, Julien Vayssière, and Gunter Bitz

Abstract With the introduction of IT to conduct business we accepted the loss of a human control step. For this reason, the introduction of new IT systems was accompanied by the development of the authorization concept. But since, in reality, there is no such thing as 100 per cent security; auditors are commissioned to examine all transactions for misconduct. Since the data exists in digital form already, it makes sense to use computer-based processes to analyse it. Such processes allow the auditor to carry out extensive checks within an acceptable timeframe and with reasonable effort. Once the algorithm has been defined, it only takes sufficient computing power to evaluate larger quantities of data. This contribution presents the state of the art for IT-based data analysis processes that can be used to identify fraudulent activities.

1 Introduction

Nowadays, it would be impossible to run a modern company without processes supported by IT, and such a company would no longer be competitive due to the enormous costs involved. This was the reason why large companies first converted their financial accounting processes to IT and then followed up with other processes. As a consequence, *Enterprise Resource Planning* (ERP) systems can today be found in innumerable companies. The implementation of IT resulted in many improvements.

Ulrich Flegel
SAP AG, Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, Germany, e-mail: ulrich.flegel@sap.com

Julien Vayssière
Smart Services CRC, Australian Technology Park, Locomotive Workshop Suite 9003, 2 Locomotive St., Eveleigh NSW 2015, Australia, e-mail: julienv@smartservicescra.com.au

Gunter Bitz
SAP AG, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany, e-mail: gunter.bitz@sap.com

Processes became faster while costs dropped. Vast paper archives became a thing of the past, giving way to IT systems that allowed rapid access to data.

At the same time, however, the introduction of IT prompted the loss of a human control step. For example, orders would leave the company electronically and fully automated, with no human intervention. For this reason, the introduction of new IT systems was accompanied by the development of the authorization concept. In practical terms, this was not a new development, but simply a question of adjusting existing IT-based access control principles to suit business processes. The signature concept was replicated in the form of the approval workflow. A system of authorization now made it possible to restrict certain tasks to certain employees in the company, and to share critical business processes between several employees.

The security of the electronic world of business stands or falls with the quality of the *Separation of Duties* (SoD) model. That is why it is of paramount importance that due care is exercised in analysing this system for risks and then minimizing them. This is, of course, also supported by software nowadays (*e.g.*, SAP GRC Access Control [12]).

But since, in reality, there is no such thing as 100 per cent security; auditors are commissioned to examine all transactions for misconduct. Given the sheer quantity of data that is processed today, it would be utterly futile to attempt to check all of it manually. It would take an entire army of auditors to check just one fiscal year's worth of data in a reasonable amount of time. Since the data exists in digital form already, it makes sense to use computer-based processes to analyse it. Such processes allow the auditor to carry out extensive checks within an acceptable timeframe and with reasonable effort. Once the algorithm has been defined, it only takes sufficient computing power to evaluate larger quantities of data.

1.1 Data Analysis Methodology

1.1.1 General

It is not always possible to establish a clear division of functions. Particularly in small and midsize companies, the personnel requirements to support a full SoD implementation would exceed reasonable levels, and would lead to a hike in administrative costs. Furthermore, the SoD implementation would have to take reserves into account to cover illness and vacation. However, since accounting and reporting tasks can be managed by considerably fewer staff than the number of roles involved in a full SoD implementation, different roles must inevitably be covered by the same person.

Generally, people are prepared to accept the higher risk factor that accompanies a partial SoD implementation for the sake of saving on administrative costs. That is why the analysis of data is particularly important in this area in order to discover fraudulent activities in the company.

In large companies also, data analysis has a twofold benefit. On the one hand, it is possible to uncover cases where employees have discovered and exploited gaps in the SoD concept, or where two or more employees have collaborated in order to deliberately bypass SoD. On the other hand, such findings serve as a basis for improving the SoD concept in order to prevent the same activities from recurring in the future. And last but not least, even the best SoD concept will always contain process steps that are to be carried out by one single employee who can find a way to act fraudulently. This is the case, for instance, with credit cards, where a further approval step would reduce the flexibility of the credit card to absurdity.

For all companies, it is therefore worthwhile to regularly analyse all available posting data. It is not of particular significance whether the data is stored in several different systems, since it is normal to work with data exports and imports anyway. In fact, it is usually necessary given that the analysis functionality is usually provided in an external tool.

1.1.2 Procedure

While data analysis is used in different ways, a standard procedure has also become established [5, 6, 7, 10]. Data is analysed in a GUI-based analysis tool (such as ACL [2], IDEA [4]) or in an advanced spreadsheet (*e.g.*, Microsoft Excel or OpenOffice Calc).

Using known fraud and corruption indicators, queries are programmed to test for occurrences of these indicators.

The results then undergo a manual plausibility check before being passed on to the auditors. This takes place because the presence of certain indicators does not constitute conclusive evidence of deliberately fraudulent behavior. Many such results are false-positives, where rules have perhaps been violated due to inadequate internal processes, but the employee did not act with fraudulent intent.

Particularly good analysis algorithms immediately conduct cross reference checks of the indicators found so that the evidence becomes stronger.

In a "scoring" procedure, points are awarded in relation to the weighting allocated to each indicator, and these points are added up to achieve different views of the data:

- Total for each employee
- Total for each business process
- Total for each project

If one of these points totals exceeds a predefined threshold, the respective employee, business process, or project undergoes a thorough audit.

2 Survey of Technology for Fraud Detection in Practice

In this section we will first have a look at the available general approaches for detecting fraud and then survey the state of the art of software tools used in practice for detecting insider fraud. We will then address the question of why and how fraud detection is different from intrusion detection at host or network level. Finally, we will list a number of challenges faced by designers of fraud detection tools.

2.1 General Approaches for Intrusion and Fraud Detection

The art and science of detecting intrusions and fraud in monitoring and transaction data has a history of over 25 years and a complete survey of the literature in this field is clearly out of the scope of this document. Approaches to intrusion detection can be classified by the kind of behavior that is modeled in order to detect intrusions. For each class a host of diverse methods for implementing the approach have been proposed in the literature, here we will only name a few that are well known. Each approach has inherent advantages and shortcomings. The following classification translates seamlessly to the domain of fraud detection.

Table 1 Desirable properties of intrusion and fraud detection methods; +: the method exhibits the desired property; -: the method does not exhibit the desired property

Desired properties	Misuse Detection	Specification-based Detection	Anomaly Detection
Low false positives, esp. for unknown behavior	+	+	-
Low false negatives, esp. for unknown behavior	-	+	+
High specificity of alarms	+	-	-
Requires no training of models	+	+	-
Requires no manual modeling	-	-	+

Employing models of intrusion behavior for intrusion detection is relatively straightforward. This approach is denoted as **misuse detection**. Known domain knowledge is modeled in a suitable framework, such as finite state machines, Petri nets and regular expressions. The stream of observed and relevant events is then matched against these models. The occurrence of a match triggers an alarm. Alarms may provide attack-specific information for mitigation, since the respective domain knowledge already exists. Since an alarm is only generated, if a model has been matched, by definition no false positives would occur. This assumes that the modeling framework is sufficiently expressive and that the models are sufficiently specific. However, since only already known attacks can be represented, unknown attacks go undetected. A continuous manual and thus costly maintenance of the model knowledge base is necessary.

The approach of **anomaly detection** assumes

1. that normal behavior, be it the behavior of a user, a protocol or process, can be accurately described, *e.g.*, statistically,
2. that attacks deviate from normal behavior, and
3. that all deviations from normal behavior represent attacks.

At design-time the normal behavior is learned from training data, which must not contain any attacks. The machine learning approaches employed to do so are many-fold: statistics, Markov processes, data mining, support vector machines, artificial neural networks, artificial immune systems, etc. During runtime, observed behavior is compared to the learned models of normal behavior and deviations are flagged as alarms. Since the approach considers deviations from normal behavior, no explicit knowledge about attacks is required and unknown attacks can be detected. Since, however, no explicit attack knowledge is used the alarms are unspecific and cannot provide further information on the attack and how to mitigate its effects. Anomaly detection systems usually come with a much higher false alarm rate than misuse detection systems due to the following reasons. The underlying assumptions do not hold entirely:

1. Modeling behavior may not be possible with the required accuracy, the machine learning methods often are inherently fuzzy and the selection of features may be specific to the given environment and may not translate to other environments. The initial training therefore is a costly process that needs to be adapted to the target environment.
2. There is no general argument why attacks would always deviate from normal behavior, and the smaller the deviations the system needs to detect, the more heavily it is prone to false alarms.
3. There is no general argument why all deviations from normal behavior would need to be attacks. Non-attack deviations result in false alarms.

The third approach is called **specification-based detection**, since deviations from an a priori defined specification of security-conforming behavior are detected. At design time a specification of allowed behavior is generated manually or automatically. Automated approaches usually work in domains with strongly limited behavior space, such as network protocols. Manually modeling allowed behavior for users is a cost-intensive task. During run-time observed events are compared to the specification and deviations are flagged as attacks. Implementation techniques leverage compiler technology, where the specification is defined as a grammar and a parser checks the input events against the grammar. Since no explicit attack knowledge is used, alarms are unspecific. However, this approach does not require initial training and comes with low false positive and false negative rates.

Table 1 summarizes the described properties of all three detection approaches.

In the domain of fraud detection, mainly anomaly detection techniques are used to flag unusual behavior that needs further manual examination and misuse detection heuristics are employed to detect known domain-specific fraud schemes.

2.2 *State of the Art of Fraud Detection Tools and Techniques*

Fraud detection tools work by analysing data already stored and processed by a number of information systems. It is important at this point in the discussion to note that most of the tools made available to fraud auditors do not go beyond importing data and distilling it for the purpose of displaying information from which the fraud auditor can extract cases of suspected fraud. These tools do not usually directly identify potential fraud cases or classify them through any sort of machine-learning technique. The classification stage is entirely left to the human. This of course explains why generic office tools (e.g., Microsoft Access and Excel) are as popular with fraud auditors as specialised packages such as ACL and IDEA.

A useful categorization criterion for tools used by fraud auditors is the degree of coupling between the fraud detection tools and the company and its existing information systems. This yields three categories of fraud detection tools: generic, specialized and custom-built tools.

2.2.1 **Generic Tools and Computer-Assisted Audit Tools**

These tools provide the user (internal auditor for example) with a toolbox for importing, processing and exporting data for the purpose of fraud detection. However, very little knowledge about what types of fraud to look for and how to detect it is built into these tools. Typical examples would be ACL (Audit Command Language) [2] and IDEA (Interactive Data Extraction and Analysis) [4] but also generic data analysis packages such as SAS or Microsoft Excel. Even if ACL and IDEA are sold as “user friendly”, they still require a certain amount of training for fraud auditors to be able to use them. When used to analyse the data extracted from an ERP system, for example, it is not uncommon to find that intimate knowledge about the data structures used in the ERP system is required in order to perform the data analysis effectively.

A study performed by the Institute of Internal Auditors in 2006 to find out which technology auditors rely on revealed that ACL was the clear leader in the field of Computer-Assisted Audit Tools (CAATS). However, this needs to be put in perspective with the fact that, when it comes to data analysis, ACL ranked second behind Microsoft Excel and in front of Microsoft Access. Clearly, CAATS do not deliver sufficient auditing-specific value to justify dropping standard tools altogether.

When asked specifically about fraud detection, users express a preference for CAATS, even though Excel and Access together represent the tool of choice for 40% of the persons interviewed.

One could also mention functionalities embedded in ERP software, such as the *Auditing Information System* (AIS) provided by SAP ERP. It is a set of reports that an auditor can run to generate the information most needed in financial audits and fraud audits. Among those reports are fraud-specific ones such as multiple invoices, one-time vendor accounts and analysis of payment terms.

2.2.2 Specialised Tools for Fraud Detection

We refer here to tools for fraud detection that are specific to a given industry or function. These tools are typically very effective at the analysis of a certain type of fraud but lack the flexibility to adapt to adjacent markets.

An example is the Triversity company that SAP acquired in 2005. Triversity is specialized in delivering software to the retail sector. Triversity tools help managers of retail chains detect fraud committed by store managers, and helps store managers to detect fraud committed by their employees.

The tool can be seen as a set of configurable reports that are run on the data collected from POS (Point of Sale) devices. This data documents everything down to a keystroke on the cash register. A manager may want, for example, to study the ratio of cash purchases over credit card purchases per checkout employee since a ratio lower than the average may be an indication of an employee pocketing cash from cash purchases. Another example would be the rate of product returns across stores since this is an indicator of staff or management creating fake product returns in order to pocket the reimbursed cash.

The insurance business also has a range of players for insurance claim fraud detection. Insurance Services Offices [9] for example, acts as a data hub for a large number of insurance companies for the purpose of detecting fraud among customers. This data can be visualized with powerful specialized tools such as NetMap Analytics [11] that help auditors relate disparate pieces of information such as claims, people, addresses and phone numbers in order to detect scams.

Specialised tools exist also, of course, in the banking sector. Being able to detect credit card fraud is a particularly challenging task since the entire fraud detection step takes place within the few seconds between the card swipe and the decision to authorize or reject a transaction. Detection techniques are a combination of customer scoring and ad-hoc rules learnt from experience about which transactions are likely to be fraudulent[8].

Anti-Money Laundering is another niche market for fraud detection tools. Even if banks do not have a direct monetary incentive to detect dirty money, there are strong regulatory frameworks in place that force them to do so. A number of tools exist and are used by most banks. However, banks do not invest much effort into correlating data between banks in order to detect the more advanced money-laundering schemes. They usually rely on passing information about suspicious transactions on to the banking regulatory authorities who may then investigate the matter on their own.

2.2.3 Custom-built Tools

Custom-built tools form the silent majority of fraud detection tools. They range from crude data processing in Excel to sophisticated Access applications or even custom developments conducted directly inside ERP systems.

The major advantage of custom-built tools is that they are uniquely fitted to the business of the company at hand. Even so, anecdotal evidence reports that it may take a couple of years to get things right. Extracting the information and lowering the rate of false positives generated by the analysis stage are the main challenges.

The predominance of Excel as a platform for custom-built tools is reinforced by the fact that Excel is a popular exportation format for data stored in enterprise applications since the spreadsheet paradigm fits the paradigm of the relational database table quite well.

It is of course difficult to provide a survey of custom-built tools since they are not usually publicized. One should not however underestimate their degree of sophistication: some of them come complete with a workflow component for auditable processing fraud alerts and email or SMS notification of new cases.

Large auditing firms are also known to develop their own tools for fraud detection. However, since the tool captures in software the differentiating advantage of the auditing firm, *i.e.*, the auditor's know-how, these tools are not shared.

There is one example of a custom-built tool that has been publicized: the Sherlock tool developed by the auditing firm *PriceWaterhouseCoopers* (PWC) [3]. This tool takes as input a set of accounting statements produced by a large number of companies. In this set are a set of documents that are known to be fraudulent. The system is trained to classify financial statements as indicative of a company that may be "cooking the books", *i.e.*, doctoring financial statements to their advantage. The objective for a large auditing firm such as PWC is obvious: improving the efficiency and the effectiveness of the audit procedure. The classifier used is a Bayesian classifier together with the expectation-maximization (EM) algorithm. A lot of other approaches for applying machine learning to fraud detection have been proposed in the research literature. Neural networks feature prominently here.

However, even though some results from the research community have managed to make their way into custom-built or specialized software, such as credit card or telecom fraud detection, they have not yet been made available through generic tools such as ACL or IDEA. The reason is most likely that applying a classification technique to a given problem in order to obtain valuable results is no "push button" software, but rather a project in itself which requires both a lot of expertise with classification techniques and solid training data. It is actually no surprise that only an auditing firm such as PWC, with thousands of customers across different industries and millions of accounting documents at hand, could effectively train a system to detect financial statement fraud.

3 Why Fraud Detection is not the Same as Intrusion Detection

A number of research attempts have been made to investigate how techniques and tools developed for the purpose of detecting intrusions on computer network or into computers at the level of the operating system could be re-used for the purpose of detecting fraud.

Even though this may be a good idea in some cases, we believe there are a sufficient number of differences between the two application domains that warrant a specific approach for research in fraud detection. Let us try to list a few:

An Intrusion Detection System (IDS) typically scans logs of network traffic or OS-level events to detect known patterns of attack. These patterns are known either because they have been observed many times in the wild, or because the particular piece of software that implements the attack (the so-called *exploit*) has been reverse-engineered. Once an exploit is released, people with little or no knowledge about the particular vulnerability that is being exploited can perform the attack.

Insider fraud detection, on the other hand, is about rare events. We are not talking about an email gateway being probed thousands of times a day for a vulnerability patched long ago, or about a Web server under attack from a worm such as Code Red. For a medium-sized company, we may be talking about an accountant putting in a fake invoice in March, a warehouse worker driving home with a brand free new TV in September and the head of purchasing favouring a high school friend in a competitive bid in exchange for a ski vacation around Christmas.

Even if insider fraud is a problem for all organisations, it is still a rare event when compared to the day-to-day operations of a company. In addition, each case of fraud is different. This is because most fraudsters are not criminal masterminds that carefully plan fraud schemes for months and exchange information in back alleys with other fraudsters before perpetrating the act. The typical fraudster may be your average employee who, under circumstances of unbearable personal financial stress (gambling, divorce, drug addiction, etc.), and having stumbled upon weaknesses in the internal controls of the company, decides to step over the line and use the trust bestowed on him by his employer for his personal gain. Fraud events are therefore rare and polymorphic, posing challenges for the misuse detection approach.

That being said, it is known to IT security experts that malware and exploit code today is polymorphic and stages multi-step attacks, to the point that naïve misuse detection approaches no longer work. This indeed brings IDS techniques closer to fraud detection. Still, we believe the variability of attacks mounted by insiders is a lot greater than for network attacks. Sadly, there is not enough data available on fraud to allow us to substantiate this claim. Ideally, we would like to study the log files for a fraudster that had been active for several years before being caught¹ to find out if he tried to vary his attacks to avoid detection or, on the other hand, if not being caught made him complacent and he repeated the same fraud pattern over and over again. The lack of fraud audit data makes it difficult to evaluate and compare fraud detection methods.

¹ According to the ACFE Report to the Nation [1] it takes on average 18 months to detect a case of fraud after the fraudster starts perpetrating fraud.

4 Challenges for Fraud Detection in Information Systems

To finish, we provide a non-exhaustive list of the challenges faced by fraud detection tools in information systems:

The move to the business level: Most tools are still meant to be used by a hybrid of internal auditor and ERP administrator. Typically, they are used by tech-savvy auditors. We need tools that speak the language of auditors and automatically map queries and results into the data structures used by ERP products. This is not as much a technical problem as a problem of usability and expressiveness of the tools.

Outsourcing: the outsourcing of many of the functions performed by a company to external service providers, possible in other countries, makes it harder to audit the part of a business process that is outsourced. This is more of a technical problem than a legal problem, since most outsourcing contracts have provisions for auditing of service providers, and the requirements of compliance frameworks such as Sarbanes-Oxley explicitly extend to service providers.

Lowering the cost of finding fraud: Many companies do not even try to detect fraud because of the upfront cost of buying and configuring software, or developing your own, and staffing the position of internal auditor. The widespread belief (confirmed by many fraud surveys) that fraud is something that only happens to others reinforces this position of denial. Our objective is therefore to lower the TCO (Total Cost of Ownership) of fraud detection tools and improve their detection capabilities such that they pay for themselves, rather than being seen as a cost that has to be bore because of regulatory requirements. This is not an inaccessible goal when one knows that, according to the American Association of Certified Fraud Examiners, companies loose on average 5% of their revenues to fraud.

5 Summary

With the introduction of IT to conduct business we accepted the loss of a human control step. For example, orders would leave the company electronically and fully automated, with no human intervention. For this reason, the introduction of new IT systems was accompanied by the development of the authorization concept. That is why it is of paramount importance that due care is exercised in analysing this system for risks and then minimizing them. This is, of course, also supported by software nowadays. But since, in reality, there is no such thing as total security; auditors are commissioned to examine all transactions for misconduct. Since the data exists in digital form already, it makes sense to use computer-based processes to analyse it. Such processes allow the auditor to carry out extensive checks within an acceptable timeframe and with reasonable effort.

The research community has come up with a number of different approaches over the years to the problem of fraud detection. However, the state of the art of fraud detection by professional auditors is very different. Most of the tools used are simple tools for data import and analysis that leave the detection and classification of potential fraud cases entirely to the human being. In a recent survey, nearly half the auditors who responded reported using generic tools such as Microsoft Excel or Microsoft Access to do their work. This is clearly a sign that the current offering of Computer-Assisted Auditing Tools (CAATS) is not sufficient, and that existing research results have failed to transfer into usable tools for auditors.

We distinguish between generic, specialized and custom-built fraud detection tools. Studying the offers in detail, we realized that the seemingly primitive state of the field can be explained by the fact that most of the advanced tools developed are either custom-built, and therefore very rarely publicized, or are used in very specialized solutions that, for commercial reasons, like to remain discrete on how exactly they achieve fraud detection. Another thing to keep in mind is that the target audience for fraud detection tools, *i.e.*, fraud auditors, cannot be described as tech-savvy. As a result, it is hard to find people with the right combination of computer science expertise and auditing expertise to match the various approaches to the actual problems faced by auditors. We argue that the detection of insider fraud is a similar, yet different problem from that of detecting intrusion at host or network level. A main reason is that attacks against networks are more automated and frequent than attempts at perpetrating fraud through enterprise information systems, which we believe show more variability. The fact remains, however, that we have very little data available to study patterns of fraud in enterprise systems. We present a number of challenges faced today by the designers of fraud detection solutions, which we hope will help steer the community in the right direction. The first challenge is to bridge the gap between existing fraud detection tools and the business level. We need tools that any auditor can use. Another challenge that appeared in the last decade is the increasing outsourcing of non-core functions of a company to external entities. Even if the legal frameworks in place extend auditing requirements to these outsourcing companies, lots of technical and sometimes legal barriers exist to cross-company fraud detection. Finally, and this is more of a business issue than strictly a research one, the enterprise applications industry needs to lower the Total Cost of Ownership of detecting fraud in enterprises. We need to come to the point when installing and using anti-fraud tools pays for itself through the money recovered.

Acknowledgements

The research leading to these results has received funding from the German Federal Ministry of Economy and Technology under promotional reference 01MQ07012 (project THESEUS/TEXO). The authors take the responsibility for their contribution.

References

1. ACFE: Report to the nation. URL <http://www.acfe.com/resources/publications.asp?copy=rtn>. Accessed on 09/21/2007.
2. ACL. URL <http://www.acl.com>. Accessed on 09/20/2007.
3. Bay, S., Kumaraswamy, K., Anderle, M.G., Kumar, R., Steier, D.M.: Large scale detection of irregularities in accounting data. In: ICDM, pp. 75–86. IEEE Computer Society (2006)
4. Caseware: IDEA. URL <http://www.caseware-idea.com>. Accessed on 09/20/2007.
5. Coderre, D.G.: Fraud Detection: A Revealing Look at Fraud, 2 edn. Ekaros Analytical (2004)
6. Coderre, D.G.: CAATTs and Other BEASTs for Auditors, 3 edn. Ekaros Analytical (2005)
7. Coderre, D.G.: Fraud Analysis Techniques Using ACL. Wiley (2009)
8. Economist: Secrets of the digital detectives. URL http://www.economist.com/displayStory.cfm?story_id=7904281. Accessed on 09/21/2007.
9. Insurance Services Offices. URL <http://www.iso.com>. Accessed on 09/20/2007.
10. Lanza, R.B.: Payables Test Set for ACL. Ekaros Analytical (2003)
11. NetMap Analytics. URL <http://www.netmap.com.au>. Accessed on 09/20/2007.
12. SAP: GRC Access Control. URL <http://www.sap.com/solutions/grc/accessandauthorization/index.epx>. Accessed on 09/21/2007.