Anomaly Detection for Soft Security in Cloud based Auditing of Accounting Systems

Mats Neovius[1] and Bob Duncan[2]
[1]*Faculty of Natural Sciences and Technology, Åbo Akademi University, Turku, Finland*
[2]*Computing Science, University of Aberdeen, Aberdeen, U.K.*

Keywords: Cloud Security, Accounting Systems Audit, Anomaly Detection.

Abstract: Achieving information security in the cloud is not a trivial exercise. When the systems involved are accounting software systems, this becomes much more challenging in the cloud, due both to the systems architecture in use, the challenges of proper configuration, and to the multiplicity of attacks that can be made against such systems. A particular issue for accounting systems concerns maintaining a proper audit trail in order that an adequate level of audit may be carried out on the accounting records contained in the system. In this paper we discuss the implications of the traditional approach to such systems and propose a complementary soft security solution relying on detecting behavioural anomalies by evidence theory. The contribution is in conceptualising the anomalies and providing a somewhat theoretical solution for a difficult and challenging problem. The proposed solution is applicable within any domain consisting of rigorous processes and risk of tampering or data exfiltration, such as the cloud based accounting systems.

1 INTRODUCTION

In previous work (Duncan and Whittington, 2016b), the authors highlighted a number of issues with defining and maintaining a proper audit trail. This problematic issue is particularly important for the audit of accounting software systems which run on the cloud. Maintaining proper accounting records is a fundamental part of the process of preparing statutory accounts for business enterprises. Where the audit trail is compromised, assurance that the records contained in the accounting system are accurate is reduced, or at worst lost. Punitive penalties (Sox, 2002)(Law, 1996)(Crown, 1998) are levied for failure to properly maintain such records, or ensure proper privacy protection of data held, thus it is vital to ensure effective steps are taken to ensure the security and privacy of all such data processed by enterprises. Next year, when the EU General Data Protection Regulation comes into force, enterprises must additionally, in the event of a data breach, be able to identify all records accessed, compromised, or deleted within a period of 72 hours. Failure to do so can result in a fine of up to 4% of global turnover. This is why we have chosen to investigate this important issue.

The systems architecture of enterprises has seen little change over past decades, with the possible exception of cloud computing. Many of the attacks against such systems are continuing to be successfully deployed against enterprises. The approach on this problem is often by using technical means alone frequently being expressed as policies authorising some action. But the business architecture of an enterprise comprises people, process and technology (PWC, 2012), not technology alone, thus such solutions are generally doomed to failure, as suggested by (Duncan et al., 2013)(Duncan and Whittington, 2014)(Duncan and Whittington, 2015d)(Duncan and Whittington, 2015b), who note such approaches ignore the impact of people and process on security.

Since the evolution of cloud, there has been some reluctance on the part of enterprises to adopt the technology. Much of this resistance stems from a lack of trust in the cloud service providers' (CSP)'s ability to maintain good levels of security and privacy (Armbrust et al., 2010). In the light of ever increasing globalization, concern over difficulties in meeting cross border data protection regulations (Kuner, 2010), also has an impact on take-up of cloud. Concern, too, has been expressed (Walden, 2011), over issues involved in trying to obtain access to records held in foreign jurisdictions. Concern has also been expressed about the difficulties of cloud audit (Duncan and Whittington, 2015c)(Duncan and Whittington, 2016a), and in

Neovius, M. and Duncan, B.
Anomaly Detection for Soft Security in Cloud based Auditing of Accounting Systems.
DOI: 10.5220/0006305504990506
In *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pages 471-478
ISBN: 978-989-758-243-1

particular the mechanics of maintaining a proper audit trail (Duncan and Whittington, 2016b).

We therefore propose a novel complementary means to address the issue of agent specific behavioural anomaly (in literature sometimes called outlier) detection in the context of audit in a cloud based system. According to Chandola et al. "anomaly detection refers to the problem of finding patterns in data that do not conform to expected behaviour" (Chandola et al., 2009). The challenges are quite well understood, with a major one being categorising the "normal region" and deciding whether or not an anomaly is of malicious character. Concerning related work, we are unaware of such algorithms for detecting behavioural anomalies in cloud based audit systems; with Doelitzscher et al. (Doelitzscher et al., 2011) noting this as future work. In later work, the same authors applied neural networks to learn a behavioural model of the user (Doelitzscher et al., 2013). This is a valid approach whenever the environment is consistent, but in an ever changing environment, such as cloud, more adaptive means are called for. Realistically, in accounting systems such a change may be due to a judicial or a personnel change.

In this paper considering evolving cloud-based systems, the anomaly detection mode is unsupervised, making the assumption that the events of normal behaviour are far more frequent than those of anomalous character. Hence, the contribution of this paper is twofold: First we introduce a discussion on the necessity of complementary security measures to policy-based ones. This is characterised especially in cloud-based systems where a customer may not be aware/responsible of/for all vulnerabilities. Realistic ones include personnel changes or migration on the CSP's part. Secondly we propose a theoretical method addressing the findings of the first contribution. This method is mathematically sound and complementary (proved in related work) and provides means for prompt reaction in case of anomalous behaviour. Realistically, the reaction may be a further approval by a higher ranking officer.

The outline of the paper is as follows: In Section 2, we first look at major vulnerabilities identified in security breach survey reports, and in Section 3, we will consider the typical accounting software architecture for an enterprise, and in Section 4, we consider the motivation for this research. In Section 5, we consider how we might approach resolving these issues, in Section 6, we outline formally our experience based theory, in Section 7 we illustrate how our approach might work, in Section 8. we discuss the possible impact this might have on security and privacy, and in Section 9, we draw our conclusions and discuss future work.

2 MAJOR ENTERPRISE BREACH VULNERABILITIES

A quick look at the annual security breach reports issued by many security companies (Verizon, 2014)(PWC, 2014)(Trustwave, 2013) clearly demonstrate the security and privacy problems still faced today, including the fact that the same attacks continue to be successful year on year, as demonstrated by the five year summary of the Verizon reports shown in Table 1 below.

Table 1: Verizon Top 5 Security Breaches — 2011-2014 (1=Highest) (Verizon, 2011)(Verizon, 2012) (Verizon, 2013)(Verizon, 2014).

Threat	2011	2012	2013	2014
Hacking	1	1	1	1
Malware	2	2	2	2
Misuse by comp. employees	4	5	5	5
Physical theft or unauth. access	3	4	3	4
Social Engine.	5	3	4	3

This raises an obvious question, why is it that the same attacks are successful year on year? And the obvious answer has to be that these vulnerabilities continue to work, year after year after year. Thus we should now study the details of a typical systems architecture for an enterprise accounting system.

3 THE TYPICAL SYSTEMS ARCHITECTURE OF AN ACCOUNTING SYSTEM

Traditionally, bookkeeping has been done carefully only by trained professionals and subject to rigorous processes. The reason is simple, the bookkeeping contains the true state of a business. Today, bookkeeping is done with accounting software generating momentary reports on the state of the business supporting the executives in their decision making. However, accounting software systems have long ceased to be just that — accounting software systems. Since the evolution of Business Process Management (BPM) and Service Oriented Architecture (SOA), especially since being enabled by continuing advances in the development of technology, enterprises have

moved away from using a traditional accounting software system to a more comprehensive Enterprise Resource Planning (ERP) model. This model comprises an additional range of enterprise management packages, such as Finance Management (FM), Purchasing/Procurement Management (PPM), Customer Relationship Management (CRM), Supply Chain Management (SCM), Business Intelligence (BI), Human Resources (HR), Distribution/Warehousing Management(DWM), Inventory Control Management (ICM), Project Management (PM),and Service Management (SM) giving the executives a realistic forecast.

Where once this would have been provided by a proprietary closed system, virtually all modern ERP systems now use the client/server paradigm. If these are hosted in-house, the business remains in control of the access (physical and logical) to the data but need to invest in hardware. An example of this would be having the hardware in a room to which only some people have access and storing the backups in a safe with an accounting software requiring a remote desktop login accepting logins only within the LAN. The bottleneck in such solutions will soon become the Internet bandwidth especially as getting ever more accurate and longer reaching forecasts, the amount of internal and external data is increasing and more sophisticated algorithms are run on this data. For this and because many enterprises operate globally, an obvious solution was to move to a web based architecture running a back end database with hardware provided on-demand, i.e. in the cloud. An obvious place to start, then, is to look at vulnerabilities found in web services. These provide the main point of contact for adversaries trying to breach the enterprise's systems. For this, we turn to the Open Web Application Security Project (OWASP) Top Ten Vulnerabilities list. OWASP is an international organisation which engages in the conception, development, acquisition, operation, and maintenance of applications that can be trusted. Every three years, they produce a Top Ten list of the most dangerous vulnerabilities.

Many of these vulnerabilities involve weaknesses in databases, their configuration and maintenance, and it is fair to suggest that all of the top ten vulnerabilities in table 2 have the potential to impact on ERP systems. While there are many defences that can be deployed to address the OWASP vulnerabilities, it is clear from the effect of the Verizon security breach reports shown in table 1 that solving the OWASP vulnerabilities alone will not fully solve the problem.

For example, social engineering attacks come in many forms. Typically one thinks of devastating phishing attacks, but also less obvious ones like revealing software versions etc., may prove to be of a critical nature. Blind hacking can be very effective when being perpetrated by a skilled actor, but when fuelled by information obtained through social engineering attacks, the results can be lethal. Thus, though social engineering attacks present (only) the third highest successful attack on enterprises, they are likely to be a great enabler for hacking.

Table 2: OWASP Top Ten Web Vulnerabilities — 2013 - 2007 (OWASP, 2013).

2013	2010	2007	Threat
1	1	2	Injection Attacks
2	3	7	Broken Authentication and Session Management
3	2	1	Cross Site Scripting (XSS)
4	4	4	Insecure Direct Object References
5	6	-	Security Misconfiguration
6	-	-	Sensitive Data Exposure
7	-	-	Missing Function Level Access Control
8	5	5	Cross Site Request Forgery (CSRF)
9	-	-	Using Components with Known Vulnerabilities
10	-	-	Unvalidated Redirects and Forwards

4 THE MOTIVATION FOR THIS RESEARCH

No matter what technological assistance or rigorous processes are complied with, this may not be enough to thwart a new form of attack. Often, the more carefully safeguarded data, the more attractive a social engineering attack gets. Examples include Stuxnet, the data published by Wikileaks and, though not categorised as an attack, the case of the Snowden revelations. Common to all are that very, very carefully guarded systems / information were compromised. Accounting systems are different in the way that data need not necessarily be exfiltrated, or a system interfered with, but mere inconsistency will prove costly both in labour to restore them, and in the consequences of possible wrong forecasts giving rise to bad decisions. Hence, mere data inconsistency could create havoc. Many enterprises are keen to achieve compliance with standards bodies, such as the International Organization for Standardization (ISO). However, the pace of evolution of new technology far outstrips the capability of international standards organisations to keep up with the changes (Willingmyre, 1997), a point well illustrated by the fact that it took 8

years for the ISO to even mention cloud in their standards. With cloud standards only now starting to filter through, it is clear that it will still be some time before a comprehensive cloud security standard will emerge. It is also clear that some enterprises might also suffer from the rapidity of technology change through being unable to keep up with such rapid change. A good point to consider with the ISO is that many global enterprises are keen to achieve compliance with ISO standards, and further, the shift from the compliance based approach to a risk based approach is very welcome.

Conventional technical solutions can help address the OWASP vulnerabilities, all of which in some way might impact on our enterprise ERP solution. What these technical solutions cannot do, however, is to tackle the major security breach issues as outlined in the Verizon breach list covered in table 1. As we stated earlier, the business architecture of an enterprise comprises a combination of people, process and technology. While technological solutions might be well advanced, and processes might be well documented, the weakest link is generally people. But even technology and processes can fall short as a consequence of the rapidity of change, as discussed above, meaning that documented technology changes might very well not be up to date. The processes of an enterprise basically document the internal information flow of all internal processes, and as stated, may not be fully up to date. The third component, people, should at least be very well documented by the HR department. Guidance from management comes in the form of Policies and Procedures. First, through the development of a policy manual for the enterprise, where management express the general policies which convey the thinking of management to guide line management. This will, of necessity, be written at a high level of abstraction. Second, procedure manuals will necessarily be written in some good level of detail, so that those responsible for carrying out process tasks understand in more depth what their responsibilities are. It is rare to find an enterprise where absolutely every element of technology, process, policies and procedures are all fully documented and up to date.

While people might be very well documented by HR, it is less likely that their training will be fully up to date. Often, a new employee is started and put to work right away, before training procedures can be fully implemented, thus putting the enterprise at an immediate disadvantage. Also, some people are not very switched on, and can be susceptible to social engineering attacks. Sadly, the security of enterprise will only ever be as good as the weakest person employed by the enterprise. No matter how up to date the documentation within an enterprise is, this may not be enough to thwart a new form of attack. Thus, it is essential that an enterprise keep their finger on the pulse with regard to evolving attacks, and take steps to mitigate these at the earliest opportunity. It is clearly of vital importance that an enterprise pay particular attention to change management, in order to ensure all documentation can be maintained properly and kept up to date. Likewise, any change in any element, should be checked through all the others to ensure there is no knock on effect which has not been considered.

Therefore, a change in technology should prompt a review of the policies and procedures manuals, to see if there is any impact there, and a review of processes carried out to ensure the existing process documentation is still fit for purpose. There should also be a check on the people of the enterprise, in order to understand whether this technology change will require more, or fewer, people to be involved in the processing tasks. This is not a one way check. Every change must be checked through to ensure no consequential impact on the other aspects of the business architecture. Given the propensity for systems architecture to be moving towards the client/server model, the database systems in use merit special mention, and this is especially important for cloud based systems. For the database, ensuring that a proper audit trail is maintained is vital. Often, cloud instances are spooled up on demand, and shut down when demand falls. If special attention has not been paid to the collection of audit trail data, which is often collected on the running instance, once it is shut down, the data collected dies with it, which means that other arrangements must be made to ensure the data is properly collected and stored. If the data is gone, then even the best forensic scientist will not be able to help. When data collected is part of the financial system of the enterprise, then it is especially important to safeguard both the data and the audit trail.

For the audit trail there are some key issues that should be considered:

1) logs must not be stored on the same host that generates them;
2) they should be append-only;
3) their originator must be identifiable. Moreover, the;
4) log's integrity must be protected, and;
5) the log is subject to confidentiality protection.

As this trail may not improve on the security level provided by technology and processes as such, it provides an insight to the past usage of the database.

This highlights some serious challenges. Often the database is hosted on the same machine as the rest of the systems. Proper security controls are rarely properly specified, meaning many unauthorised users can have access to logs. And should a malicious user gain access to the system, and be able to escalate privileges, then there would be nothing to stop them from modifying logs, or deleting them altogether. Another concern springs from the people employed by the enterprise. If they use the systems improperly through misunderstanding, laziness, lack of training, or through malicious intent, then this could give rise to a serious issue. Indeed, where an inside user works in collusion with an external malicious agent, this presents one of the greatest dangers to an enterprise. Yet another concern surrounds the attitude of management. Duncan and Whittington (Duncan and Whittington, 2015a) have warned of the dangers present when management fail to take security seriously. Where this is the case, we do recognise that this is sometimes slow to change. In total, the traditional technology providing policy-based means to improve security yearns for complementary means to follow up on the agent's behaviour. This paper introduces a novel means that looks at the overall business architecture of the enterprise by detecting behavioural anomalies. The method would utilise the audit trail to (learn) construct a behavioural normality for an agent. In case this normality is broken (anomaly is recognised), the actions are domain specific. Realistically this could suggest a secondary approval by a manager. We now turn to address how detecting anomalous behaviour might be effected.

5 DETECTING ANOMALOUS BEHAVIOUR

Related work on anomaly detection by tracking behavioural patterns for soft security include that of Device Comfort (Marsh et al., 2011). Device Comfort has been developed to detect a mobile device's level of comfort on its user. It considers a situational comfort level to be the device ownership combined with the context (location, task, etc.) where anomalies are actions causing the situational comfort level to drop under some given threshold level. Later research has targeted the mobile device to "know" its user's diurnal patterns in order to detect anomaly behavioural patterns (Bicakci et al., 2014). This is used to detect, for example, anomalies in the location and use of the mobile device indicating a possible theft, giving rise to further action based on an "anomaly score" that remains undefined.

Characterising for accounting systems means however that their operational purpose remains unchanged and all activity is exclusive and exhaustive. A significant change in an agent's functional access to the system is thereby a sign of anomaly. In cases when the agent's work duties have changed, this can however be the "new normal" requiring the system to adapt. In this paper we do present a method responding to these challenges. We do effectively the same as with Device Comfort but for an audit system with a focus on detecting temporal behavioural anomalies instead of analysing the spatial context in which they occur. The reason is obvious, as the spatial context is the cloud, lacking any conventional spatial context. For this we use experience based theory on exclusive and exhaustive events to create a window of normal behaviour in all a system's interactions (dimensions). We apply a factor of ageing providing the essentials for adaption. Mathematically similar approaches have been used for finding a level of belief and certainty on aggregated information (Neovius, 2015). In the next Section, we describe how the experience based theory might be developed.

6 THE EXPERIENCE BASED THEORY

Consider an experience defined as a four tuple $(\delta, \varepsilon_i, \zeta, \eta)$ inspired by Teacy et al. (Teacy et al., 2006), and Krukow's (Krukow, 2006) general model augmented by a score. Here δ represent the agents, ε the datum indexed by i (typically time), ζ the tasks (events) and η the score of this experience. The history of such experiences is a set $(\delta, \varepsilon_i, \zeta, \eta)$.

The observing entity, in the context of this paper, is always the audit system. For the sake of readability, let the exclusive and exhaustive tasks on the audit trail ζ = (n, r, e) where n = 'new entry', r = 'read', and e = 'edit an old entry'. Three exclusive and exhaustive outcomes are spanned by an equilateral triangle; with each additional outcome increasing one dimension on the plane. Thus, with four outcomes, the faces span a tetrahedron. Moreover, consider atomic behaviour, where an agent conducts at most 1 task at any given time.

Adding a new experience to the history is straight forward: $(\delta, \varepsilon_0, \zeta, \eta) \cup \{(\delta, \varepsilon_i, \zeta, \eta)\}$ where $0 \leq i$. Thus, the experience based behavioural view may be retroactively constructed from the audit trail. On the history, a projection provides a situational view on some spanned plane, i.e. $\exp(usr_1, \varepsilon_j, x, \eta) = (\varepsilon_j, \eta)$. This projection provides all $x \subseteq \zeta$ entries of agent(s) *usr1* within the timespan denoted by {j}. In addition,

aggregation of dimensions to a binomial view is easy as for the exclusiveness of the tasks by letting x denote a non-empty set of dimensions with $\bar{x} = \zeta \setminus x$. Moreover, assume for the sake of this example the score η indicating the task's level of severity. Also assume that the initiating agent is unique and known and that it takes full responsibility for the task.

Having a history of experiences, any agents' (one or a group of agents) temporal behaviour can be examined over any time span or cycle as defined by the domain expert. For this, a decay constant $0 \leq \lambda \leq 1$ is implemented where the closer the constant is to 1, the less the decay. For an hourly or daily behaviour, temporally older experiences reasonably weigh less than recent, hence at ε_m the decay d is $d_{\varepsilon_m}(\exp(\delta, \varepsilon_j, \zeta, \eta)) = \{(\delta, \varepsilon_j, \zeta, \lambda^{\varepsilon_m - \varepsilon_j} * \eta)\}$. This decay provides the "trend" of the current behaviour. However, the current behaviour is not applicable on cyclic (repetitive) behaviour such as weekly, quarterly and yearly time frames. On these, we propose inversing the decay $0 \leq \theta \leq 1$ where the closer to 0, the more weight on the early readings, i.e. at ε_m the cyclic c is $c_{\varepsilon_m}(\exp(\delta, \varepsilon_j, \zeta, \eta)) = \{(\delta, \varepsilon_j, \zeta, \theta^{\lambda^{\varepsilon_m - \varepsilon_j}} * \eta)\}$. Obviously, to make this effective, the time span is the recurring pattern's time span. This enables a reference window calling for domain specific extreme cautiousness in defining the parameters.

With the decay providing the current trend and the inverse, an anomalous behaviour is defined with a set of rules on thresholds within the domain of discourse. Optimally these thresholds are defined by a function on the experiences in an event over a period of time with respect to a comparable period. For example, consider X to be a time window right now, and Y to be all events over a comparable period in the past, and let $\alpha(Y) = \sum_{j \in Y} d_{\varepsilon_X}(\exp(\delta, \varepsilon_j, n, \eta))$ providing a score indicating the new entries during that frame. To give an example, a Boolean anomaly behaviour with thresholds $0.8 \leq x \leq 1.15$ is then identified if the predicate $0.8 * \alpha(Y) \leq \sum_{j \in X} d_{\varepsilon_x}(\exp(\delta, \varepsilon_j, n, \eta)) \leq 1.15 * \alpha(Y)$ does not hold, i.e. if the momentary situation X deviates too heavily from the reference period Y (yesterday, last week...). If this is the case, further domain specific actions may be required, or an alert raised, e.g. approval of an agent with more rights. Obviously, an ensemble of complementary reference periods can be defined and included in the predicate as the domain specialist wishes.

7 ILLUSTRATING THE APPROACH

This section will illustrate the approach described in Section 6. The example used above included three exclusive and exhaustive tasks where anomalous behaviour is indicated by the level of "out of the ordinary". For illustration, we need an n-dimensional barycentric co-ordinate system where n is the cardinality of ζ, here 3 ($\{n,r,e\}$) and thus a tetrahedron. For presentation purposes, we coarsen the 3-dimensional space to 2-dimensions where x = n and $\bar{x} = \{r,e\}$. This gives us a 2 dimensional barycentric space, i.e., a triangle (one face of the tetrahedron), and a way of illustrating, including the transformation formulae (below), originally brought forward by Jøsang (Jøsang, 1997). Moreover, let $\sum_{j \in X} d_{\varepsilon_m}(\exp(\delta, \varepsilon_j, \zeta, \eta)) = (1,3.6,0.06)$ for respective $\zeta = (n,r,e)$ and for the comparable period $0.9,3.2,0.01$. Let the normalisation of these values include a constant W for "*do not know*" with a given value 2 where higher values require more experiences for certainty. Then the values for each x is $\sum x / (\sum x + \sum \bar{x} + W)$. The vectors on a proposition are defined:

$$\overrightarrow{b}(x) = \sum_{x_i \subseteq x} \overrightarrow{b}(x_i) \text{ and } \overrightarrow{b}(\bar{x}) = \sum_{x_i \subseteq \bar{x}} \overrightarrow{b}(x_i).$$

In the two-dimensional space, these two vectors span the behaviour right now, and are vector-valued functions on the ζ with range $[0, 1]^n$. They are subadditive, as of the W, as illustrated in Figure 1 below. The parameter a_x is the base-rate parameter indicating the expected development.

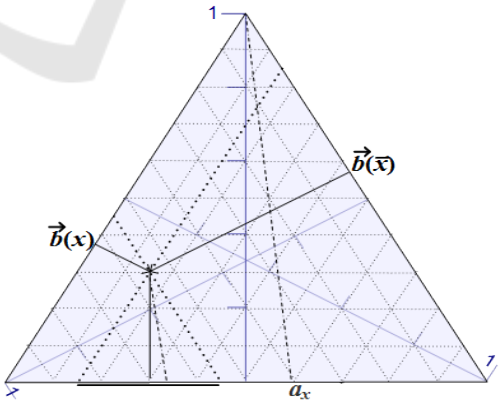

Figure 1: Situational behaviour.

The comparable periods centroid is calculated similarly using the cyclic delay. The tolerance however spans an oval around this centroid, in this case 0.8 and 1.15 indicating the "normal" acceptable behavioural space of the agent. This is illustrated in Fig-

ure 2 below. Of note is that every time a window spans an area of its own, where decisions on framing rules of how many of these may be broken without being tagged as anomalous behaviour, must be taken by the domain expert. In a perfect world of consistent behaviour the time windows would span the same area.

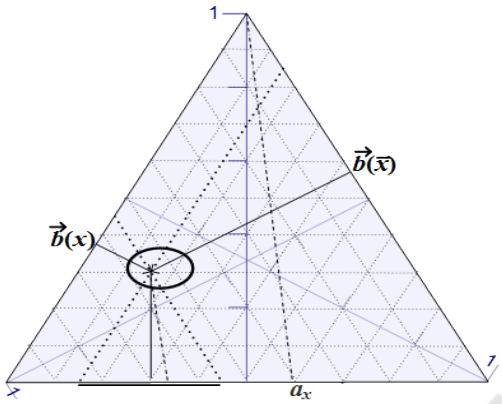

Figure 2: Comparable time window's tolerance and situational behaviour within this.

The interpretation of the method is such, that the reference period defines a point on which a level of tolerance is put. If the current window deviates from the reference period, the behaviour is anomalous. In Figure 2, the oval spans the reference window with tolerance and the dot the current behaviour. If the dot slides outside the oval, the current behaviour is anomalous with respect to its reference period. In the next Section, we discuss various points which will require clarification in the use of this system.

8 DISCUSSION

There are a number of points that need to be clarified in the use of our proposed system. Every new user will be highlighted on first access to the system, since there will be no historic data to compare against current activities. This should trigger a need for a responsible person to investigate whether this activity comes from a properly authorised user. If necessary, a time or event count delay could be placed on processing of their input until their actions are duly authorised. An alternative would be to set up a set of default expected behaviour whenever a new employee starts with the enterprise. Thus any entries resulting from a malicious user would instantly stand out as suspect. Strict limits on the level of access granted to new users should be in force until an established pattern of behaviour is developed. For this, other types of machine learning approaches could be used. However, as we considered a system already running, this is not the prime focus of this paper.

Moreover, situations of anomalous behaviour that are not of a malicious nature may be caused by a change of duties of a user within the enterprise. At the very least, such behaviour should be flagged up, validated and approved. As we have already stated, reliance on the system alone would be foolhardy. Rather, the system should be able to create useful alerts to highlight anomalous behaviour as it happens, so that it might be investigated before it becomes a major issue. In effect, we are proposing a highly efficient "management by exception approach", rather than either getting bogged down by too much manual checking, or worse, by doing nothing at all. The same method could be implemented as a service by network providers to users. In the final Section, we discuss our conclusions and consider future work

9 CONCLUSIONS AND FUTURE WORK

Achieving information security in the cloud calls for new means of addressing security. Porting the traditional approach developed for static client-server or mainframe system to the cloud is not possible. Problems relate to the very foundations of SaaS and the cloud paradigm, including elasticity, spatial independence etc. Moreover, as the fundamental operations such as accounting systems of enterprises are run on such foundations, the stakes are high. This sensitive information attracts fraud and calls for means to detect such attacks as soon as they happen. For this, we have in this paper proposed an algorithm detecting anomalies in the usage of the system. Stakeholders on this include the enterprise as well as the SaaS provider. This provides 'soft' security measures and complements policy based security actions, such as encryption, redundancy and many more.

The trend of IT systems today is that they become ever more cloud based, whether this be IaaS, SaaS or 'serverless computing'. For the enterprise to move the operationally critical software to a place maintained by hard core professionals is appealing. When systems are run by hard core professionals, they are better updated and less prone to cyber physical attacks. Thus, the trend has been, and is likely to continue, that fraudsters will utilise social attacks. To countermeasure such social attacks, soft security measures such as is proposed in this paper can address these issues. As future work, we intend to validate this method on a real-life benchmarking dataset.

The anomaly detection method could be run by the

enterprise or by the CSP. We chose accounting systems as the application because of the readily available audit trail and the characteristic of having to be adaptive to changes in agent.

This is the first step to a successful spoofing attack, where an intruder acts as a legitimate provider acting as the man-in-the-middle.

REFERENCES

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., and Rabkin, A. (2010). A View of Cloud Computing: Clearing the clouds away from the true potential and obstacles posed by this computing capability. *Commun. ACM*, 53(4):50–58.

Bicakci, M. V., Esfandiari, B., and Marsh, S. (2014). Anomaly detection for mobile device comfort. In *IFIP International Conference on Trust Management*, pages 93–108. Springer.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Comput. Surv.*, 41(3):15:1—-15:58.

Crown (1998). Data Protection Act.

Doelitzscher, F., Knahl, M., Reich, C., and Clarke, N. (2013). Anomaly Detection In IaaS Clouds. In *CloudCom*, pages 387–394.

Doelitzscher, F., Reich, C., Knahl, M., and Clarke, N. (2011). An Autonomous Agent Based Incident Detection System for Cloud Environments. In *Cloud Comput. Technol. Sci. (CloudCom), 2011 IEEE Third Int. Conf.*, pages 197–204.

Duncan, B., Pym, D. J., and Whittington, M. (2013). Developing a Conceptual Framework for Cloud Security Assurance. In *Cloud Comput. Technol. Sci. (CloudCom), 2013 IEEE 5th Int. Conf. (Volume 2)*, Bristol. IEEE.

Duncan, B. and Whittington, M. (2014). Compliance with Standards, Assurance and Audit: Does this Equal Security? In *Proc. 7th Int. Conf. Secur. Inf. Networks*, pages 77–84, Glasgow. ACM.

Duncan, B. and Whittington, M. (2015a). Company Management Approaches Stewardship or Agency: Which Promotes Better Security in Cloud Ecosystems? In *Cloud Comput. 2015*, pages 154–159, Nice. IEEE.

Duncan, B. and Whittington, M. (2015b). Information Security in the Cloud: Should We be Using a Different Approach? In *2015 IEEE 7th Int. Conf. Cloud Comput. Technol. Sci.*, Vancouver.

Duncan, B. and Whittington, M. (2015c). Reflecting on whether checklists can tick the box for cloud security. In *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, volume 2015-Febru, pages 805–810, Singapore. IEEE.

Duncan, B. and Whittington, M. (2015d). The Importance of Proper Measurement for a Cloud Security Assurance Model. In *2015 IEEE 7th Int. Conf. Cloud Comput. Technol. Sci.*, Vancouver.

Duncan, B. and Whittington, M. (2016a). Enhancing Cloud Security and Privacy: The Cloud Audit Problem. In *Submitt. to Cloud Comput. 2016*, Rome.

Duncan, B. and Whittington, M. (2016b). Enhancing Cloud Security and Privacy: The Power and the Weakness of the Audit Trail. In *Cloud Comput. 2016*, Rome. IEEE.

Jøsang, A. (1997). Artificial reasoning with subjective logic. In *Proceedings of the second Australian workshop on commonsense reasoning*, volume 48, page 34. Citeseer.

Krukow, K. (2006). Towards a theory of trust for the global ubiquitous computer. *Brics.Dk*.

Kuner, C. (2010). Regulation of Transborder Data Flows under Data Protection and Privacy Law: Past, Present, and Future. *Leg. Stud.*, (016):1–90.

Law, P. (1996). Health Insurance Portability and Accountability Act of 1996.

Marsh, S., Briggs, P., El-Khatib, K., Esfandiari, B., and Stewart, J. A. (2011). Defining and Investigating Device Comfort. *J. Inf. Process.*, 19(7):231–252.

Neovius, M. (2015). Adaptive experience-based composition of continuously changing quality of context. In *The Seventh International Conference on Adaptive and Self-Adaptive Systems and Applications*, page 21 26. IARIA.

OWASP (2013). OWASP Top Ten Vulnerabilities 2013.

PWC (2012). UK Information Security Breaches Survey - Technical Report 2012. Technical Report April, PWC2012.

PWC (2014). 2014 Information Security Breaches Survey: Technical Report. Technical report.

Sox (2002). Sarbanes-Oxley Act of 2002.

Teacy, W. L., Patel, J., Jennings, N. R., and Luck, M. (2006). Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198.

Trustwave (2013). Trustwave Global Security Report. Technical report.

Verizon (2011). 2011 Data Breach Investigation Repeort: A study conducted by the Verizon RISK Team in cooperation with the United States Secret Service and Others. Technical report.

Verizon (2012). 2012 Data Breach Investigation Report: A study conducted by the Verizon RISK Team in cooperation with the United States Secret Service and Others. Technical report.

Verizon (2013). 2013 Data Breach Investigation Report: A study conducted by the Verizon RISK Team in cooperation with the United States Secret Service and Others. Technical report.

Verizon (2014). 2014 Data Breach Investigations Report. Technical report.

Walden, I. (2011). Law Enforcement Access in a Cloud Environment. *Leg. Stud.*, (74):1–19.

Willingmyre, G. T. (1997). Standards at the Crossroads. *StandardView*, 5(4):190–194.